UNIVERSIDAD CARLOS III DE MADRID
Departamento de Teoría de la Señal y Comunicaciones

DOCTORAL THESIS

NONLINEAR POPULATION MONTE CARLO METHODS FOR BAYESIAN INFERENCE

Author: EUGENIA KOBLENTS LAPTEVA
Supervised by: JOAQUÍN MÍGUEZ ARENAS
March 2015

Tesis Doctoral:	NONLINEAR POPULATION MONTE CARLO METHODS FOR BAYESIAN INFERENCE
Autor:	Eugenia Koblents Lapteva
Director:	D. Joaquín Míguez Arenas
Fecha:	

Tribunal

Presidente:

Vocal:

Secretario:

Agradecimientos

Ahora que termina este largo trabajo, quiero dar las gracias a todas las personas que han contribuido de alguna manera a que hoy esté escribiendo esto.

Por supuesto, quiero agradecer a mi director de tesis Joaquín su paciencia, ayuda e implicación en este trabajo. Por haber hecho posible que esto saliera adelante, dándome la confianza y la flexibilidad que necesitaba. Por haberme pemitido trabajar a mi ritmo y a mi manera.

A mi pequeña familia, David, Irene y Daniel, por hacerme feliz cada día. Vuestro cariño me ha dado la energía y las ganas de seguir adelante con este proyecto.

También quiero agradecer a mis dos grandes familias por formar una parte tan importante de mi vida, por ser mi mayor apoyo y mi mayor fuente de satisfacciones. Muy en especial, quiero dar un enorme gracias a mi madre y a mis suegros, por el papel fundamental que han tenido en el desarrollo de esta tesis. Estoy convencida de que sin vuestra ayuda incondicional esto no hubiera sido posible.

Agradezco también a mis amigos, anteriores y posteriores al comienzo de este trabajo, por haberme acompañado en estos largos años de tesis. Por ayudarme y escucharme cuando lo he necesitado, y por entenderme como sólo vosotros sabéis.

Nunca olvidaré esta etapa tan especial que han sido mis años de doctorado, y que me han aportado tantísimo en todos los aspectos de la vida. Gracias a todos los que habéis formado parte de ello. Gracias a vosotros no sólo he podido realizar este trabajo sino que he disfrutado con ello, quizá más importante.

Abstract

In the present work we address the problem of Monte Carlo approximation of posterior probability distributions and associated integrals in the Bayesian framework. In particular, we investigate a technique known as population Monte Carlo (PMC), which is based on an iterative importance sampling (IS) approach. The PMC method displays important advantages over the widely used family of Markov chain Monte Carlo (MCMC) algorithms. Opposite to MCMC methods, the PMC algorithm yields independent samples, allows for a simpler parallel implementation and does not require a convergence period. However, both IS and PMC suffer from the well known problem of degeneracy of the importance weights (IWs), which is closely related to the curse of dimensionality, and limits their applicability in large-scale practical problems.

In this thesis we present a novel family of PMC algorithms which specifically addresses the degeneracy problem arising in high dimensional problems. In particular, we propose to perform nonlinear transformations to the IWs in order to smooth their variations and increase the efficiency of the underlying IS procedure, specially when drawing from proposal functions which are poorly adapted to the true posterior. This technique, termed nonlinear PMC (NPMC), avoids the need for a careful selection of the proposal distribution and can be applied in fairly general settings. We propose a basic NPMC algorithm with a multivariate Gaussian proposal distribution, which is better suited for unimodal target distributions. For general multimodal target distributions, we propose a nonlinear extension of the mixture PMC (MPMC) algorithm, termed adaptive nonlinear MPMC (NMPMC) method, which constructs the importance functions as mixtures of kernels. Additionally, the new technique incorporates an adaptation step for the number of mixture components, which provides valuable information about the target distribution.

We also introduce a particle NPMC (PNPMC) algorithm for offline Bayesian inference in state-space models, which allows to approximate the posterior distribution of both the model parameters and the hidden states given a set of observed data. A major difficulty associated to this problem is that the likelihood function becomes intractable in general nonlinear, non-Gaussian state-space models. To overcome this drawback, the new technique resorts to a particle filter (PF) approximation of the likelihood, in a manner equivalent to the widely used particle MCMC (PMCMC) algorithm. All the

proposed algorithms are described in Chapter 3.

In Chapter 4 we provide a convergence analysis of the nonlinear IS (NIS) technique which is at the core of the proposed NPMC inference algorithms. We investigate the error introduced by two types of nonlinear transformations of the IWs, termed tempering and clipping. We also account for the additional error introduced by the weight approximation obtained with a PF. We provide explicit upper bounds for the errors incurred when approximating integrals of bounded functions using the NIS technique.

Through Chapters 5, 6 and 7 we numerically assess the performance of the proposed techniques and compare them to state of the art algorithms. In Chapter 5 we present some simple simulation examples which illustrate the principle behind NPMC and NMPMC and the performance improvement attained by the NIS technique. As a first practical application, in Chapter 6 we have considered the popular (and challenging) problem of estimating the rate parameters and the hidden states in a stochastic kinetic model (SKM). SKMs are highly multivariate systems that model molecular interactions in biological and chemical problems. We have applied the proposed PNPMC algorithm to this problem and performed an extensive simulation comparison with the powerful PMCMC method. In Chapter 7 we address the problem of Bayesian parameter estimation in α-stable distributions, which allow to describe heavy-tailed and asymmetric data. In this last application example, we provide simulation results both with synthetic and real data.

Resumen

En este trabajo hemos abordado el problema de la aproximación de distribuciones *a posteriori*, e integrales con respecto a éstas, mediante métodos de Monte Carlo. En concreto, nos hemos centrado en una técnica conocida como *population Monte Carlo* (PMC), que está basada en un enfoque de muestreo enfatizado (*importance sampling*, IS) iterativo. El método PMC presenta importantes ventajas frente a la familia de métodos de Monte Carlo basados en cadenas de Markov (*Markov chain Monte Carlo*, MCMC). Al contrario que los algoritmos MCMC, el método PMC permite generar muestras independientes de la distribución de interés, admite una implementación paralelizada y no requiere establecer períodos de convergencia. Sin embargo, tanto el método IS como el PMC sufren el conocido problema de degeneración de los pesos, que está muy relacionado con la *maldición de la dimensión* y limita su aplicabilidad en problemas prácticos de alta complejidad.

En esta tesis doctoral presentamos una nueva familia de algoritmos PMC que aborda de manera específica el problema de la degeneración de los pesos en alta dimensión. Concretamente, proponemos realizar transformaciones no lineales a los pesos para suavizar sus variaciones e incrementar la eficiencia del proceso de IS, en particular cuando la función de importancia no se ajusta bien a la distribución *a posteriori* de interés. La técnica propuesta, llamada PMC no lineal (*nonlinear* PMC, NPMC), no requiere una selección cuidadosa de la función de importancia y se puede aplicar en gran variedad de problemas. Proponemos un esquema NPMC básico que emplea una función de importancia Gaussiana, que es más adecuada para aproximar distribuciones unimodales. Para el caso general de distribuciones *a posteriori* multimodales, proponemos una extensión no lineal del algoritmo *mixture* PMC (MPMC), que denominamos MPMC no lineal adaptativo (*nonlinear* MPMC, NMPMC), que construye las funciones de importancia como mezclas de distribuciones núcleo. Además, el método propuesto incorpora un paso de adaptación del número de componentes de la mezcla, lo cual proporciona una valiosa información acerca de la distribución objetivo.

También proponemos un algoritmo llamado *particle* NPMC (PNPMC) para inferencia Bayesiana *offline* en modelos de espacio de estados, que permite aproximar distribuciones *a posteriori* tanto de los parámetros fijos del modelo como de la secuencia de estados ocultos, en base a una secuencia de observaciones. La principal dificultad en esta clase de problemas es

que la función de verosimilitud no se puede evaluar de forma exacta en modelos de espacio de estados no lineales y/o no Gaussianos. Para afrontar esta limitación, el algoritmo propuesto recurre a una aproximación de la verosimilitud mediante filtrado de partículas (*particle filtering*, PF), de manera equivalente al ampliamente usado algoritmo de *particle* MCMC (PMCMC). Los algoritmos propuestos se describen en el Capítulo 3.

El Capítulo 4 presenta un análisis de convergencia de la técnica de muestreo enfatizado no lineal (*nonlinear* IS, NIS). Hemos investigado el error de aproximación introducido por dos tipos de transformación no lineal en los pesos, denominados *tempering* (suavizado) y *clipping* (recorte). También analizamos el error adicional introducido por la aproximación de los pesos obtenida mediante PF. En todos los casos, proporcionamos cotas explícitas para el error de aproximación obtenido mediante la técnica de NIS.

A lo largo de los Capítulos 5, 6 y 7, evaluamos numéricamente las prestaciones de los algoritmos propuestos y los comparamos a otros algoritmos existentes en la literatura. En el Capítulo 5 presentamos algunos ejemplos sencillos que ilustran los principios básicos de los métodos NPMC y NMPMC y la mejora en el rendimiento introducida por la técnica de NIS. Como primera aplicación práctica, en el Capítulo 6 hemos considerado el popular y complejo problema de la estimación de parámetros y poblaciones en modelos estocásticos cinéticos (*stochastic kinetic models*, SKMs). Los SKMs son sistemas de alta dimensión que modelan las interaciones moleculares que ocurren en problemas biológicos y químicos. Hemos aplicado el algoritmo PNPMC propuesto a este problema y hemos realizado una comparación exhaustiva con el algoritmo PMCMC. Por otro lado, en el Capítulo 7 abordamos el problema de estimación de parámetros en distribuciones α-estables, que permiten modelar datos asimétricos y de colas pesadas. En este último caso, mostramos resultados de simulaciones realizadas tanto con datos sintéticos como reales.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter we introduce and motivate the fundamental problems addressed in this dissertation. Section 1.1 provides a brief introduction to Monte Carlo methods and discusses their use in Bayesian estimation. In Section 1.2 we outline the main contributions of the thesis. Finally, in Section 1.3 we describe the organization of the dissertation.

1.1 Bayesian inference and Monte Carlo methods

This thesis addresses the problem of estimating a set of unknown parameters, that describe a nonlinear statistical model, based on a set of measured data. Real problems in modern science and engineering often involve dealing with nonlinear and high-dimensional systems, where both the dimension of the parameters and the observations can be potentially large [38, 55, 70, 114].

In particular, we are interested in the Bayesian approach to this inference problem, which not only aims at computing point estimates of the parameters of interest, but also at providing information on the uncertainty about those values, represented in terms of their probability distribution [146, 108, 63]. In the Bayesian framework, the model parameters are assumed to be random variables themselves, endowed with a prior probability distribution that characterizes the uncertainty on their values before any data are collected. Once the observations become available, the prior distribution of the parameters is combined with their likelihood (the conditional distribution of the observations) to yield the so-called posterior distribution. This posterior distribution, opposite to conventional point estimates, provides complete information of the random parameters and allows the computation of expected values, expected errors,

etc. However, the Bayesian approach often results in high dimensional integration problems, which usually cannot be solved analytically, unless dealing with standard well-known distributions [147, 146]. Traditional deterministic approximations of integrals (such as the Riemann sum) become intractable as the dimension of the problem increases. For this reason, the development of efficient numerical methods to approximate posterior distributions in high-dimensional spaces, and integrals with respect to (w.r.t.) them, has been a very active area of research in the last decades [147, 114, 38].

A very common strategy which has been successfully applied in a broad variety of complex problems is the Monte Carlo methodology [126, 147, 62]. Monte Carlo, or simulation-based, methods are a broad family of computational algorithms which use randomly generated samples to solve numerical problems [147]. The Monte Carlo method was invented by mathematicians Stanislaw Ulam and John von Neumann and physicist Nicholas Metropolis, while working on the nuclear weapons program at the Los Alamos National Laboratory, during the World War II [126]. The name comes from the Monte Carlo casino in Monaco, because of the similarity of the technique to the act of playing roulette. This kind of methods are usually applied in high-dimensional spaces, when deterministic analytical solutions become intractable, and which often correspond to problems of great interest in practical applications [62, 114, 82, 70].

The main applications of Monte Carlo methods include simulation from probability distributions, numerical optimization and integration [147]. In this work we focus on the Monte Carlo integration problem, which allows to approximate a distribution of interest, often called a *target* distribution, and definite integrals over complicated domains. Such integration problems often arise in the Bayesian framework, where the target distribution is the posterior distribution of a set of random variables of interest, given some observed data. Instead of evaluating the integrand at a regular grid, Monte Carlo methods consider randomly generated grids of points in the space of interest. This is particularly useful in high-dimensional spaces, where the probability mass is usually concentrated in a very small (but hard to identify) region.

In this work we are concerned with two Bayesian inference scenarios. On one hand, we address the estimation of the static parameters of a probabilistic model based on a set of observations available beforehand. For this kind of problems, the natural choice is to apply an offline (batch) Monte Carlo method [147]. On the other hand, we consider the problem of inferring the hidden state and/or the static parameters in generic nonlinear

and non-Gaussian state-space models, based on a set of observations that arrive sequentially in time [12, 55]. The family of sequential Monte Carlo (SMC) methods is naturally suited to this kind of applications [55, 31, 42]. In the next sections we briefly review these two broad families of algorithms, emphasizing those more closely related to the techniques developed in this thesis.

1.1.1 Batch Monte Carlo methods

Let us consider the approximation of the posterior distribution of a set of static model parameters given some measured data. This is a very general problem that is often referred to as Bayesian model inference [108, 63, 38]. The basic Monte Carlo method tackles this problem by generating samples from the target distribution, if possible, and approximating the integrals of interest by sample means [147, 62]. However, in many practical cases it is not possible to draw from the target distribution directly and we need to resort to more sophisticated techniques. A powerful tool to draw samples from arbitrary target distributions is the family of Markov chain Monte Carlo (MCMC) algorithms [77, 66, 147]. MCMC methods aim at constructing a Markov chain whose equilibrium or stationary distribution is the desired posterior distribution [147]. Once the chain has converged, the obtained samples can be used to approximate the distribution of interest and related integrals.

A huge number of algorithms based on the MCMC principle have been proposed in the literature during the last decades. Some of the more relevant are the Metropolis-Hastings algorithm [125, 77], the Gibbs sampler [64, 34], the slice sampler [131], multiple-try Metropolis algorithm [116], reversible jump MCMC method [75] or the hybrid Monte Carlo scheme [58], to cite just a few.

MCMC methods have been applied to solve numerical problems in many and very diverse practical applications [66] in engineering [5, 70], finance [40, 82] or systems biology [161], among other fields. However, MCMC algorithms present a set of important drawbacks, that hinder their application in many practical scenarios. To be specific, the samples in the chain are produced strictly sequentially, which leaves little room for parallelization compared to other Monte Carlo methods [147, 85, 105]. Additionally, MCMC methods yield correlated samples, which implies that some thinning procedure is often needed to produce better estimates [147, 66]. Finally, the chain converges asymptotically to the target distribution, so its first elements have to be discarded (this is often termed the burn-in

period of the chain).

A common approach to overcome these difficulties is the importance sampling (IS) methodology [123, 55], which allows to perform inference on a target probability density function (pdf) based on samples generated from a proposal pdf, or importance function, and their associated importance weights (IWs). In this work we focus on a technique known as population Monte Carlo (PMC), which is based on an iterative IS approach [32, 30]. The main advantages of the PMC scheme, compared to the well established MCMC methodology, are the possibility of developing parallel implementations, the sample independence and the fact that an asymptotically unbiased estimate is provided at each iteration, which avoids the need of a burn-in period. On the other hand, an important drawback of the IS approach, and particularly of PMC, is that its performance heavily depends on the choice of the proposal distribution. When the target pdf is very narrow w.r.t. the proposal (this occurs when, e.g., the dimension of the variables of interest or the number of observations is high), the vast majority of the IWs become practically zero, leading to an extremely low number of representative samples [101, 57, 19]. This problem is commonly known as weight degeneracy and is closely related to the curse of dimensionality [19]. The issue was already mentioned in the original paper where the PMC algorithm was proposed [32]. However, to the best of our knowledge, it has not been successfully addressed in the PMC framework to this date.

Rather than facing the degeneracy issue, the effort in the field of PMC algorithms has been directed toward the design of efficient proposal functions. For instance, the recently proposed mixture PMC (MPMC) technique [30] models the importance functions as mixtures of kernels. This method is a generalization of the D-kernel PMC (DPMC) algorithm proposed in [52, 53]. In the MPMC method, the weights and the parameters of each mixture component are adapted along the iterations to minimize the Kullback-Leiber divergence (KLD) between the target density and the proposal. This scheme also suffers from degeneracy and the authors of [30] propose to apply a Rao-Blackwellization (RB) [35, 147] scheme in order to mitigate this drawback.

In the multiple marginalized PMC (MultiMPMC) algorithm [28, 149] the conditionally linear parameters are marginalized out, which allows to reduce the computational cost of the PMC algorithm [28]. Additionally, the potentially multidimensional space of interest is partitioned into several subspaces of lower dimension and handled by parallel PMC filters [149]. In [78] a PMC algorithm for the joint model selection and parameter estimation is introduced, based on a two-stage sampling procedure.

Another recently proposed PMC scheme is based on the Gibbs sampling method [50, 150] and aims at sampling efficiently from high-dimensional proposals. The authors of [50] propose to construct the proposal distributions as products of alternating conditionals, where sampling from each conditional is easy. This technique allows for an efficient sampling and evaluation procedure. However, the IWs still present severe degeneracy due to the extreme values of the likelihood function in high-dimensional spaces. Other population-based algorithms for static inference exist [85].

In general, PMC and MCMC methods for Bayesian inference applications require that the likelihood function can be evaluated up to a proportionality constant. Unfortunately, the likelihood evaluation can be very costly or even intractable in many practical scenarios of great interest in science and engineering [16, 155, 138]. As an alternative to standard model-based statistical methods, likelihood-free or approximate Bayesian computation (ABC) [143, 155] techniques allow to perform inference using forward simulation from the observation model. In the basic ABC scheme [143, 16], known as rejection ABC algorithm, the simulated and the observed data are compared in terms of some distance function, which is usually defined based on a set of summary statistics. Samples with a small distance to the observations are accepted, while the rest are rejected. The ABC approach has been combined with MCMC [122], SMC [151] and PMC algorithms [15].

1.1.2 Sequential Monte Carlo methods

In this section we focus on the approximation of the joint posterior distribution of the parameters and the hidden states in general state-space models. This is a well known problem with a broad scope. Multiple SMC methods have been applied to this problem in a wide variety of applications, including engineering, economics and biology; see [54, 55, 31, 84, 83].

When the model parameters are known, the problem is relatively simple and the various versions of the particle filter (PF) enable the recursive online estimation of the hidden states given the observations [73, 92, 115, 57]. The PF is a SMC algorithm based on a sequential IS approach, which, at each time step, generates a set of samples from a proposal distribution and computes the corresponding IWs. The obtained set of samples and weights allows for the recursive approximation of the filtering distribution of interest. Given that the PF algorithm is based on IS, it also suffers from weight degeneracy. A way of alleviating this problem is to include a so-called *resampling* step, which probabilistically removes samples with low weights

and replicates those with higher weights, allowing for a rejuvenation of the population [51].

The simplest form of the PF is known as the bootstrap filter [73], which uses the prior transition density as a proposal and performs resampling at each time step. Multiple variations of this algorithm have been proposed in the literature, such as the auxiliary PF [142], the Rao-Blackwellized PF [57, 56, 39], the Gaussian PF [102], the unscented PF [156], etc.

The problem of the joint estimation of the model parameters and the hidden states is much more challenging, because the marginal likelihood of the parameters cannot usually be computed analytically in general state-space models [42]. A popular technique to solve this problem is the powerful particle MCMC (PMCMC) algorithm of [6]. The PMCMC method resorts to a likelihood approximation computed by means of a PF, which is used to compute the acceptance probability of a Markov chain. Note that this algorithm is not a sequential one, but provides a batch solution. Additionally, it has a potentially very high computational complexity, as one needs to run a full-blown PF for each element in the chain.

The SMC square (SMC^2) algorithm proposed in [42] is a sequential offline algorithm for the joint estimation of parameters and hidden states in state-space models. This method has a structure of nested PFs and it allows for a sequential update of the posterior distribution of interest as new observations become available.

While the original SMC^2 algorithm is a sequential procedure, it is not a truly recursive algorithm, but a batch method that shares several features with the PMCMC approach [42]. A recursive version of the SMC^2, built upon a similar structure of nested PFs, but better suited for online inference, has been recently proposed in [44]. Other SMC methods for parameter estimation in state-space models have been proposed, e.g., in [93, 113, 7, 152]. A review can be found in [88, 89].

Many other SMC methods have been proposed in the literature for the estimation of static model parameters, in the kind of problems traditionally addressed by MCMC methods or other batch techniques [41, 130, 47, 32]. These methods introduce an artificial sequence of intermediate posterior distributions, which may result in an increased efficiency or present some other advantages. For example, the SMC sampler of [47] allows for the approximation of a sequence of probability distributions, both in batch and sequential settings, and includes as particular cases some relevant methods considered in this work [32, 130, 41].

Multiple combinations of SMC methods and the batch techniques reviewed in this section have also been proposed: SMC with MCMC moves

[106], adaptive and sequential MCMC [4], adaptive direction sampling [68], SMC without likelihoods [151], adaptive SMC-ABC [48], etc.

1.1.3 Practical applications

The Bayesian inference approach and the Monte Carlo methodology have been jointly and successfully applied in many complex problems in science and engineering [62, 148, 25, 110, 120, 70]. For example, it is of increasing interest in the biological sciences to develop new techniques that allow for the efficient estimation of the parameters governing the behavior of complex autoregulatory networks, as well as the populations of the interacting species [161, 72, 160]. Also numerous applications in population genetics [76], molecular biology [143], ecology and epidemiology involve dealing with high-dimensional systems [133, 120] and require the use of efficient computational techniques.

In the design of forecast techniques in meteorology and oceanography, scientists often deal with complex models involving a large number of parameters and high-dimensional observations, which require vast computational resources [3, 59, 110, 165]. Also in medical sciences many applications of interest exist: Bayesian analysis of medical time series [18], diagnostic imaging [117], medical physics [148], or risk analysis for human health [29], among many others.

Monte Carlo methods are also very important in computational physics and related fields [22]. For example, the estimation of parameters and Monte Carlo simulation of stochastic processes in astrophysics is an active area of research [112, 129, 154, 141, 119].

The Monte Carlo methodology has also been widely applied in financial and business applications [82, 70], for example to forecast how returns and prices vary over time [25, 84], for security pricing [24] or risk management [79, 159].

Finally, multiple problems in engineering involve dealing with high-dimensional data, such as location, navigation and tracking problems [1, 9, 11, 13, 153], whose aim is to estimate the position and velocity of a moving target. Among signal processing applications, image analysis [163] and speech processing [157] are some examples. In machine learning, MCMC algorithms lie at the core of many state of the art methods [5].

Despite the computational complexity of these techniques, the fast advances in the development of powerful computers in the last years have enabled Monte Carlo methods to provide solutions to many practical problems that were intractable in the past. The main difficulty often

encountered when tackling this kind of problems is the design of numerical inference algorithms that scale up efficiently with the dimension of the parameter and the observation space. The goal of this thesis is to devise and assess efficient computational Monte Carlo algorithms for the approximation of probability distributions and related integrals in high-dimensional or otherwise complex spaces.

1.2 Contributions

In this thesis we have focused on the family of batch PMC methods, both for the Bayesian estimation of parameters in static models and the joint estimation of the parameters and the hidden states in dynamical state-space models. We introduce a modification of the standard IS approach, which we call nonlinear IS, whose goal is to alleviate the inefficiency of IS in complex problems. Additionally, we propose a novel family of nonlinear PMC algorithms, which explicitly address the weight degeneracy problem of the underlying IS approach. The main contributions of this thesis can be summarized as:

- the design of a family of nonlinear PMC algorithms that present increased efficiency w.r.t. widely used state of the art techniques;
- the explicit calculation of convergence rates for a nonlinear IS scheme, that lies at the core of the proposed PMC algorithms, with either exact or approximate weights;
- the numerical assessment of the proposed algorithms in practical applications, including synthetic and real data.

In the following sections we discuss each of these contributions with some detail.

1.2.1 Proposed algorithms

In order to mitigate the degeneracy problem of standard IS schemes, we introduce a simple modification of the classical IS scheme, termed nonlinear IS (NIS), which consists in applying nonlinear transformations to the IWs in order to reduce their variations and obtain a sufficient number of effective samples. We discuss two particular types of nonlinear transformations of the IWs, based on tempering and clipping procedures.

A nonlinear PMC algorithm is readily obtained based on the NIS technique. The basic NPMC algorithm constructs the proposal distribution

at each iteration as a multivariate Gaussian distribution with moments matched to the previous sample set. This construction of the proposal is particularly suitable for the approximation of simple unimodal target distributions.

For a more general case where the target distribution can present multiple modes, we propose an extension of the MPMC algorithm, termed nonlinear MPMC (NMPMC), which updates the proposal distribution based on the transformed IWs (TIWs). This modification of the algorithm significantly increases its efficiency in complex problems. Additionally, we propose to introduce an adaptation step for the number of mixture components of the proposal pdf, which provides valuable information about the number of components required to adequately represent the target distribution.

In many practical applications of interest, the unknown parameters of a statistical model are related to the available observations by means of a state-space model. In this case the likelihood function cannot often be evaluated exactly but can, however, be approximated using a PF. We propose a particle NPMC (PNPMC) algorithm that allows to estimate fixed parameters and hidden states in state-space models resorting to a PF approximation of the likelihood function, in similar vein as the PMCMC algorithm.

1.2.2 Convergence analysis

The nonlinear transformation of the IWs in NIS, as well as the approximations in the evaluation of the standard IWs, introduce a distortion in the obtained approximations of integrals. To assess the effect of this distortion we analyze the approximation errors and provide asymptotic convergence rates for the NIS method with the tempering and clipping transformations.

Our analysis accounts for the use of an arbitrary approximation of the IWs with a clipping transformation. Additionally, we address explicitly the approximation of the IWs produced by a PF and quantify the distortion this approximation brings into the global error. In particular, we prove that the approximate integrals computed via NIS converge in L_2 (as the number of samples increases) and calculate explicit convergence rates.

1.2.3 Simulation examples and practical applications

To illustrate the performance of the proposed algorithms we consider several simulation scenarios of different complexity. We use a simple Gaussian

mixture model (GMM) to numerically evaluate the problem of degeneracy of the IWs, to illustrate the idea behind the NPMC algorithm and to compare its performance to some standard techniques. To assess the performance of the nonlinear version of the MPMC algorithm and for comparison with the original method we use a multidimensional GMM and a banana-shaped target distribution, which often arises in cosmological applications [164, 91].

As a first practical application, we have chosen the challenging problem of estimating the parameters in stochastic kinetic models (SKMs) [160, 161, 72, 128]. SKMs describe the time evolution of the population of a number of chemical species, which evolve according to a set of constant rate parameters. We show numerical results for the simple predator-prey model [158], and for the more complex prokaryotic autoregulatory model [160]. In this scenario, we compare the performance of the NPMC algorithm to the state of the art PMCMC technique.

Finally, we apply the NPMC algorithm with approximate weights to the problem of estimating the hidden parameters of α-stable distributions, whose pdf cannot be evaluated exactly. We provide simulation results with synthetic data to evaluate the accuracy of the proposed method and to compare it to alternative frequentist and Bayesian methods. Additionally, we apply the NPMC algorithm to a set of real fish displacement data.

1.3 Organization of the thesis

This document is structured into eight chapters. The first chapter is the present Introduction.

Chapter 2 contains the background material required to understand the rest of the dissertation. Firstly, we describe the Bayesian approach to statistical inference, in static and dynamical settings. Then we introduce the Monte Carlo methodology and its application to Bayesian inference. In the following sections we review the main families of Monte Carlo methods: sequential Monte Carlo, Markov chain Monte Carlo, population Monte Carlo and approximate Bayesian computation.

In Chapter 3 we introduce the inference algorithms proposed in this thesis. We discuss the degeneracy problem arising in standard IS schemes and describe a novel nonlinear IS technique that addresses this problem. Later on, we introduce a basic nonlinear PMC algorithm (which uses Gaussian proposals) and a nonlinear extension of the MPMC algorithm which includes an adaptation mechanism for the number of mixture components. In the next section we particularize the NPMC algorithm for

the approximation of posterior distributions in state-space models, based on a PF approximation of the likelihood function. Finally we discuss the connection of the proposed NPMC algorithms to relevant existing techniques and conclude the chapter with some final remarks.

Chapter 4 contains the convergence analysis of the proposed NIS technique. We first analyze the convergence properties of the tempering transformation of the IWs. Then we focus on the clipping transformation of the IWs and we provide upper bounds for the approximation error induced by the exact and approximate evaluation of the IWs. We also address the problem of approximating the IWs by means of a PF, which introduces an additional distortion in the desired approximations. We prove convergence of the NIS approximation in L_2 and calculate explicit convergence rates.

In Chapter 5 we present some preliminary simulation results that illustrate the performance of the proposed NPMC and nonlinear MPMC algorithms, applied to synthetic problems of different complexities. In the first section we use a simple GMM model to show the consequences of the degeneracy problem, occurring even in very simple examples. We assess the performance of the basic NPMC algorithm and compare it to alternative Monte Carlo methods. In the following two sections we evaluate and compare the performance of the original and nonlinear MPMC algorithms and show how the nonlinear version yields much better performance.

Chapter 6 is devoted to the practical application of the PNPMC algorithm to the estimation of rate parameters and hidden populations in SKMs. In the first section we introduce some background material on SKMs. Then, we describe the addressed inference problem and review the main existing techniques. In the following section we apply the PNPMC algorithm to a simple SKM known as predator-prey model. Then we provide simulation results for the more challenging prokaryotic model.

In Chapter 7 we apply the NPMC algorithm to the estimation of parameters of α-stable distributions. In the first section we introduce the basic concepts of α-stable distributions. Next we describe and discuss the proposed algorithm for the estimation of parameters. In the following section we provide extensive simulation results of the NPMC algorithm and the main previously existing techniques, in a synthetic simulation setting. We provide numerical results with real data corresponding to fish displacement in their habitat, which is modeled by an α-stable distribution.

Finally, in Chapter 8 we summarize the main findings of this work and propose some possible future research lines.

Chapter 2

Monte Carlo methods for Bayesian inference

In this chapter we provide the theoretical framework on which the rest of the dissertation builds up, as well as some notation used through the document. Firstly, we describe the Bayesian inference framework for the approximation of posterior distributions in static and dynamical models. Next, we introduce the Monte Carlo methodology and the importance sampling technique, which is at the core of this work. In the remaining sections we present the main families of Monte Carlo methods related to the problem addressed here, such as sequential Monte Carlo, Markov chain Monte Carlo, population Monte Carlo and approximate Bayesian computation algorithms.

2.1 Notation

We denote column vectors and matrices using boldface lower-case and upper-case letters respectively, e.g., $\boldsymbol{\theta}$, $\mathbf{y}$, $\boldsymbol{\Sigma}$. We use $\mathbb{R}^K$, with integer $K \geq 1$, to denote the set of K-dimensional vectors with real entries. A target pdf is denoted as π, a proposal density as q and the rest of pdfs as p. We write conditional pdfs as $p(\mathbf{y}|\boldsymbol{\theta})$, and joint densities as $p(\boldsymbol{\theta}) = p(\theta_1, \ldots, \theta_K)$. This is an argument-wise notation, hence $p(\theta_1)$ denotes the distribution of θ_1, possibly different from $p(\theta_2)$. A sample from the distribution of the random vector $\boldsymbol{\theta}$ is denoted by $\boldsymbol{\theta}^{(i)}$. Sets of M samples $\{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(M)}\}$ are denoted as $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{M}$. $\delta_{\boldsymbol{\theta}^{(i)}}(d\boldsymbol{\theta})$ is the unit delta measure located at $\boldsymbol{\theta} = \boldsymbol{\theta}^{(i)}$.

2.2 Bayesian inference

In this section we describe the Bayesian inference framework for different statistical models. In Section 2.2.1 we address the problem of inferring the static posterior pdf of a set of unknown parameters of a statistical model. In Section 2.2.2 we focus on a sequential setting, in which the pdf of interest corresponds to the sequence of posterior distributions encountered in dynamical models.

2.2.1 Bayesian inference for static models

The aim of Bayesian statistics in static setups is to infer the probability distribution of the unknown parameters of a statistical model from a set of observed data [63, 108]. To be specific, let $\boldsymbol{\theta} = [\theta_1, \dots, \theta_K]^\top$ be a column vector of K unobserved real random variables, which represent the parameters of a given model. In the Bayesian approach, a *prior distribution* with density $p(\boldsymbol{\theta})$ is specified, which represents the prior knowledge available on the model parameters $\boldsymbol{\theta}$ before observing any data. It is common to select the prior distribution as a well-known and tractable distribution that allows for easy evaluation and sampling procedures. The prior distribution constraints the parameter space, avoiding unfeasible solutions and helping to obtain reasonable estimates. It also allows to penalize complex models w.r.t. simpler ones. If there is no prior knowledge of $\boldsymbol{\theta}$, an uninformative prior can be used setting $p(\boldsymbol{\theta}) \propto 1$ [63].

On the other hand, let $\mathbf{y} = [\mathbf{y}_1^\top, \dots, \mathbf{y}_N^\top]^\top$ be a vector of N real random, but fixed, observation vectors, $\mathbf{y}_n \in \mathbb{R}^D$, related to $\boldsymbol{\theta}$ by way of a conditional pdf or likelihood function $p(\mathbf{y}|\boldsymbol{\theta})$. The likelihood is a function of the parameters $\boldsymbol{\theta}$ that quantifies how likely it is that the parameters have originated the observed data. However, it is not a density on $\boldsymbol{\theta}$, but on $\mathbf{y}$. For simple models, the likelihood function can often be computed analytically. However, in more complex models, it is usually impossible or computationally intractable to evaluate the likelihood exactly, and some approximation is required [6, 147, 155].

The prior distribution of $\boldsymbol{\theta}$ is combined with the information provided by the observed data $\mathbf{y}$ by means of the Bayes theorem [63, 147], yielding a so-called *posterior distribution* of the model parameters, with pdf $p(\boldsymbol{\theta}|\mathbf{y})$, given the observations. To be specific, the Bayes theorem states that

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}},$$

where $p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}) = p(\mathbf{y}, \boldsymbol{\theta})$ is the joint pdf of $\mathbf{y}$ and $\boldsymbol{\theta}$. The normalizing constant $p(\mathbf{y}) = \int p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$ is the marginal density of $\mathbf{y}$, also known as the model evidence [63]. In general, the evidence is difficult to compute and some approximate methods must be applied [147, 108]. However, when the posterior distribution is only required to be known up to a normalizing constant, the computation of the evidence can be avoided [63].

In this work we address the problem of approximating the unnormalized posterior probability distribution of $\boldsymbol{\theta}$, i.e., the (conditional) distribution with density

$$p(\boldsymbol{\theta}|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}). \tag{2.1}$$

Additionally, we are interested in computing approximations of any moments of $p(\boldsymbol{\theta}|\mathbf{y})$, i.e., expectations of the form

$$E_{p(\boldsymbol{\theta}|\mathbf{y})}[f(\boldsymbol{\theta})] = \int f(\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta},$$

where $f : \mathbb{R}^K \to \mathbb{R}$ is some real integrable function of $\boldsymbol{\theta}$.

The Bayesian approach was regarded as computationally intractable for a long time, since it often requires the computation of complex integrals in high dimension. Such computations can sometimes be avoided, namely when the prior and posterior distributions belong to the same conjugate family [63]. Nevertheless, the use of conjugate priors implies a restriction on the modeling of the available prior information and the likelihood function and limits the usefulness of the Bayesian approach as a method of statistical inference [147].

Point estimates

The posterior distribution yields all the information required about the random variable of interest. However, we are often interested in some point estimates which are representative values of the posterior distribution. For example, the posterior mean corresponds to the minimum mean square error (MMSE) estimate of $\boldsymbol{\theta}$ and can be obtained as

$$\hat{\boldsymbol{\theta}}_{MMSE} = \arg\min_{\hat{\boldsymbol{\theta}}} E[\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|^2] = E_{p(\boldsymbol{\theta}|\mathbf{y})}[\boldsymbol{\theta}] = \int \boldsymbol{\theta}p(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}. \tag{2.2}$$

The maximum a posteriori (MAP) estimator is obtained as the mode of the posterior distribution and is, in turn, given by

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{y}) = \arg\max_{\boldsymbol{\theta}} p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}),$$

where the troublesome normalizing term of the posterior distribution, $p(\mathbf{y})$, is avoided, since it does not depend on $\boldsymbol{\theta}$. The MAP estimator provides valuable information in some cases but can be misleading when the distribution of interest is very asymmetric.

If the prior distribution is uniform or uninformative, the MAP criterion reduces to finding the maximum likelihood (ML) estimate of $\boldsymbol{\theta}$, which is defined as

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta}} p(\mathbf{y}|\boldsymbol{\theta}),$$

i.e., it is obtained as the parameter vector that maximizes the likelihood function. Since the ML approach does not rely on a prior distribution over the parameters, it can often lead to overfitting and poor generalization of the model, and thus lacks all the benefits of Bayesian learning [147].

In Figure 2.1 we illustrate the Bayesian inference approach with an example. The prior pdf of a parameter θ, the likelihood function and the corresponding posterior pdf are shown, together with the MMSE, MAP and ML estimates.

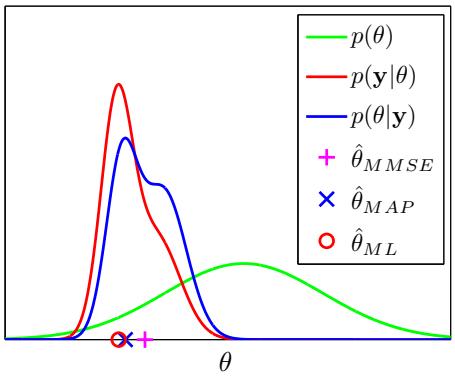

Figure 2.1: Densities and point estimates in an example of the Bayesian inference approach.

The ML approach is usually associated with optimization problems, while the Bayesian approach often results in integration problems, for example, to compute moments or marginals of a distribution of interest. Deterministic numerical methods exist that provide solutions for the described optimization and integration problems. For example, gradient-based methods are widely used for optimization, while Riemann integration allows to compute definite integrals [147, 63]. These standard numerical

methods often outperform the simulation methods when dealing with regular functions in low dimension. However, the curse of dimensionality hinders the application of deterministic techniques in high-dimensional problems [147, 38]. For example, standard numerical integration methods do not take into account that some search regions may have very low probability. Similarly, traditional optimization methods based on gradient-descent are very sensitive to multiple modes in the distribution of interest. Fortunately, the development of Monte Carlo methods for numerical integration (and optimization), together with ever faster computing devices, has paved the way for the practical application of the Bayesian (and ML) approach to complex high-dimensional inference problems.

2.2.2 Bayesian inference for state-space models

In this section we describe a different Bayesian inference problem, in which the static, yet random, parameter vector $\boldsymbol{\theta}$ is related to a sequence of observations $\mathbf{y}_n$ by way of a corresponding sequence of hidden states $\mathbf{x}_n \in \mathbb{R}^V$, $n = 0, \ldots, N$ [12, 55, 108]. This class of models, usually referred to as state-space models, allows to describe generic, possibly nonlinear, dynamic systems that often arise in practical applications in science and engineering [62, 13, 1], finance [82, 24, 84] or systems biology [120, 161]. A state-space model is defined in terms of the random sequences [12, 54]

$$\begin{cases} \mathbf{x}_n \sim p(\mathbf{x}_n|\mathbf{x}_{n-1}, \boldsymbol{\theta}) & \text{(transition equation)}, \\ \mathbf{y}_n \sim p(\mathbf{y}_n|\mathbf{x}_n) & \text{(observation equation)}, \end{cases} \tag{2.3}$$

for $n = 1, \ldots, N$, where $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \boldsymbol{\theta})$ is a transition pdf and $p(\mathbf{y}_n|\mathbf{x}_n)$ is the likelihood function at time n.

These equations imply that hidden states and data can in general be generated by nonlinear functions of the state and some noise disturbances. The initial state $\mathbf{x}_0$ has prior distribution $p(\mathbf{x}_0)$. We use $\mathbf{x}_{0:n} = [\mathbf{x}_0^\top, \ldots, \mathbf{x}_n^\top]^\top$ and $\mathbf{y}_{1:n} = [\mathbf{y}_1^\top, \ldots, \mathbf{y}_n^\top]^\top$ to denote the state and observation sequences up to time n, respectively. Additionally, we denote by $\mathbf{x} = \mathbf{x}_{0:N}$ and $\mathbf{y} = \mathbf{y}_{1:N}$ the complete sequence of hidden states and observations, respectively. Such models can be viewed as "missing data" models, since the hidden states $\mathbf{x}_n$ are not observed [101, 36, 61]. This kind of nonlinear dynamical systems are straightforward to simulate given the Markovian assumptions [147]. Additionally, the conditional density of the state and the observations can be factorized as

$$p(\mathbf{x}|\boldsymbol{\theta}) = p(\mathbf{x}_0) \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{x}_{n-1}, \boldsymbol{\theta}) \quad \text{and} \quad p(\mathbf{y}|\mathbf{x}) = \prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{x}_n),$$

respectively.

The aim of a statistical inference method for the described state-space model is the computation of the posterior distribution of a collection of state variables $\mathbf{x}$ and/or of the parameters $\boldsymbol{\theta}$, conditioned on a set of observations $\mathbf{y}$. This is a broad class of problems, which includes as specific cases filtering, smoothing or prediction of the hidden state, estimation of fixed parameters or joint estimation of both the state and the parameters [54, 57, 42, 12]. In this work, we are interested mainly in Bayesian filtering and the joint estimation of the parameters $\boldsymbol{\theta}$ and the hidden state $\mathbf{x}$.

Bayesian filtering with known parameters

Assuming the model parameters $\boldsymbol{\theta}$ are known, the aim of Bayesian filtering [55, 12] is to compute recursively in time the posterior filtering density $p(\mathbf{x}_n|\mathbf{y}_{1:n}, \boldsymbol{\theta})$, $n = 1, \ldots, N$, given by

$$p(\mathbf{x}_n|\mathbf{y}_{1:n}, \boldsymbol{\theta}) = \frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{1:n-1}, \boldsymbol{\theta})}{p(\mathbf{y}_n|\mathbf{y}_{1:n-1}, \boldsymbol{\theta})} \propto p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{1:n-1}, \boldsymbol{\theta}), \quad (2.4)$$

where $p(\mathbf{x}_n|\mathbf{y}_{1:n-1}, \boldsymbol{\theta})$ is the predictive density of the state at time n given the observations up to time $n-1$, i.e.,

$$p(\mathbf{x}_n|\mathbf{y}_{1:n-1}, \boldsymbol{\theta}) = \int p(\mathbf{x}_n|\mathbf{x}_{n-1}, \boldsymbol{\theta})p(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1}, \boldsymbol{\theta})d\mathbf{x}_{n-1} \quad (2.5)$$

and $p(\mathbf{y}_n|\mathbf{y}_{1:n-1}, \boldsymbol{\theta})$ is a normalizing constant independent of $\mathbf{x}_n$, of the form

$$p(\mathbf{y}_n|\mathbf{y}_{1:n-1}, \boldsymbol{\theta}) = \int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{1:n-1}, \boldsymbol{\theta})d\mathbf{x}_n. \quad (2.6)$$

Equations (2.5) and (2.4) represent the prediction and update steps, respectively, which form the basis of the optimal Bayesian solution to the recursive filtering problem [12]. The filtering posterior distribution can be computed exactly in the case of linear and Gaussian models by means of the Kalman filter [87, 2]. Extensions of the Kalman filter exist for nonlinear models (e.g., the extended Kalman filter [2], the unscented Kalman filter [86] and others), but they often yield poor performance [145]. Monte Carlo methods are powerful alternative strategies to handle more general dynamical systems [54, 145].

Bayesian inference with unknown parameters

If the model parameters $\boldsymbol{\theta}$ are unknown, the Bayesian filtering problem becomes more involved, as the parameters need to be calibrated from the

data. If observations are collected one at a time and parameters have to be estimated sequentially together with the hidden state, the inference problem is known as online parameter and state estimation. In this case the joint posterior pdf is given by

$$p(\boldsymbol{\theta}, \mathbf{x}_{0:n}|\mathbf{y}_{1:n}) = \frac{p(\mathbf{y}_{1:n}|\mathbf{x}_{0:n})p(\mathbf{x}_{0:n}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{y}_{1:n})}, \quad n = 1, \dots, N, \tag{2.7}$$

and it has the filtering density $p(\boldsymbol{\theta}, \mathbf{x}_n|\mathbf{y}_{1:n})$ as a marginal.

On the contrary, if the batch of observed data is available beforehand, the estimation of the parameters and the hidden states can be performed offline using the whole set of observations $\mathbf{y} = [\mathbf{y}_1^\top, \dots, \mathbf{y}_N^\top]^\top$. In this case, the joint posterior density is given by

$$p(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}), \tag{2.8}$$

where $\mathbf{x} = [\mathbf{x}_0^\top, \dots, \mathbf{x}_N^\top]^\top$. If our interest restricts to the model parameters, the marginal posterior density of the parameters is given by

$$p(\boldsymbol{\theta}|\mathbf{y}) = \int p(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})d\mathbf{x} \propto p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}),$$

where the likelihood function of $\boldsymbol{\theta}$ is

$$p(\mathbf{y}|\boldsymbol{\theta}) = \int p(\mathbf{y}|\mathbf{x})p(\mathbf{x}|\boldsymbol{\theta})d\mathbf{x} = E_{p(\mathbf{x}|\boldsymbol{\theta})}[p(\mathbf{y}|\mathbf{x})]. \tag{2.9}$$

The marginal likelihood $p(\mathbf{y}|\boldsymbol{\theta})$ is a very important quantity in Bayesian inference in state-space models and many existing methods require the possibility of evaluating or approximating this value accurately. Moreover, the likelihood of the parameters is the normalizing constant of the posterior density of the hidden state given $\boldsymbol{\theta}$, namely, $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}) = p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})/p(\mathbf{y}|\boldsymbol{\theta})$.

Inferring the model parameters thus requires integrating over all the possible realizations of the hidden states, or missing data, $\mathbf{x}$, which are usually high-dimensional. These integrals cannot be computed analytically in a general nonlinear, non-Gaussian model. However, Monte Carlo methods allow to accurately approximate them in many cases [147, 42, 6]. Once an estimate of the parameters is obtained, the filtering task to infer the hidden state can be performed as in equations (2.4) and (2.5).

2.3 Monte Carlo methods

In this section we present the basics of Monte Carlo integration applied to static target probability distributions of the type described in Section

2.2.1. However, the same concepts and techniques can be equally applied to sequential setups described by a state-space model, as long as the length of the sequences $\mathbf{x}_n$ and $\mathbf{y}_n$ is finite. We also introduce the importance sampling methodology [147, 123], which is the basis of the algorithms developed in this work, as well as its main practical drawback, the degeneracy of the IWs [101, 19]. Finally, we review the main families of Monte Carlo methods used in this work.

2.3.1 Monte Carlo integration

The basic Monte Carlo integration method [147] aims at approximating integrals of the form

$$(f, \pi) = \int f(\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta} = E_{\pi(\boldsymbol{\theta})}[f(\boldsymbol{\theta})], \tag{2.10}$$

where f is a real, integrable function of $\boldsymbol{\theta}$ and $\pi(\boldsymbol{\theta})$ is some pdf of interest (often termed the *target* density). In the Bayesian framework described in Section 2.2.1, the target density is the posterior pdf of $\boldsymbol{\theta}$, i.e., $\pi(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y})$. If $\pi(\boldsymbol{\theta})$ is some standard pdf, then it is straightforward to draw a random i.i.d. (independent and identically distributed) sample $\Theta^M = \{\boldsymbol{\theta}^{(i)}\}_{i=1}^M$ from $\pi(\boldsymbol{\theta})$ and then build a random discrete measure

$$\hat{\pi}^M(d\boldsymbol{\theta}) = \frac{1}{M}\sum_{i=1}^{M} \delta_{\boldsymbol{\theta}^{(i)}}(d\boldsymbol{\theta}),$$

where $\delta_{\boldsymbol{\theta}^{(i)}}(d\boldsymbol{\theta})$ is the unit delta measure located at $\boldsymbol{\theta} = \boldsymbol{\theta}^{(i)}$. Based on the sample Θ^M we can also readily obtain an approximation $(f, \hat{\pi}^M)$ of (f, π), namely

$$(f, \pi) \approx \left(f, \hat{\pi}^M\right) = \int f(\boldsymbol{\theta})\hat{\pi}^M(d\boldsymbol{\theta}) = \frac{1}{M}\sum_{i=1}^{M} f(\boldsymbol{\theta}^{(i)}). \tag{2.11}$$

This approximation almost surely (a.s.) converges to (f, π) (as M goes to infinity) by the law of large numbers [46, 147, 162]

$$\lim_{M\to\infty} \left(f, \hat{\pi}^M\right) = (f, \pi) \quad \text{a.s.}$$

The central limit theorem [46, 162] implies, in addition, that

$$\sqrt{M}[(f, \pi) - (f, \hat{\pi}^M)] \to \mathcal{N}(0, \sigma^2) \tag{2.12}$$

in distribution, as $M \to \infty$, where $\mathcal{N}(0, \sigma^2)$ denotes a Gaussian distribution with 0 mean and variance σ^2. Equation (2.12) reveals that the Monte

Carlo approximation error decays at a rate of $1/\sqrt{M}$, independently of the dimension K. This fact favors the application of the Monte Carlo methodology in high-dimensional problems, opposite to traditional deterministic approaches, such as the Riemann approximation [147]. This approach is known as exact Monte Carlo but it can only be applied when it is possible to sample from the target distribution directly, which is often not the case in practice.

As an example of application of the exact Monte Carlo method, we consider the evaluation of the likelihood function $p(\mathbf{y}|\boldsymbol{\theta})$ given in equation (2.9). In principle, it is possible to approximate this integral as an average of the likelihoods $p(\mathbf{y}|\mathbf{x}^{(i)})$ over a set $\{\mathbf{x}^{(i)}\}_{i=1}^{M}$ of exact Monte Carlo samples from the density $p(\mathbf{x}|\boldsymbol{\theta})$, that is,

$$p(\mathbf{y}|\boldsymbol{\theta}) = E_{p(\mathbf{x}|\boldsymbol{\theta})}[p(\mathbf{y}|\mathbf{x})] \approx \frac{1}{M}\sum_{i=1}^{M} p(\mathbf{y}|\mathbf{x}^{(i)}), \quad \text{where } \mathbf{x}^{(i)} \sim p(\mathbf{x}|\boldsymbol{\theta}). \quad (2.13)$$

This approach, however, is often computationally intractable, because it demands drawing a huge number of samples M to obtain a useful approximation of the posterior $p(\mathbf{y}|\boldsymbol{\theta})$, since the probability of generating a state realization $\mathbf{x}^{(i)}$ similar to the observations can be extremely low [101].

2.3.2 Importance sampling

A common approach to overcome the problems of the basic Monte Carlo procedure is to apply an importance sampling (IS) methodology [123, 147]. IS is based on an alternative representation of equation (2.10) of the form

$$(f, \pi) = \int f(\boldsymbol{\theta}) \frac{\pi(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} q(\boldsymbol{\theta}) d\boldsymbol{\theta},$$

where $q(\boldsymbol{\theta})$ is the so-called proposal distribution or importance function. The key idea is to generate an i.i.d. sample of size M, $\Theta^M = \{\boldsymbol{\theta}^{(i)}\}_{i=1}^{M}$, from the (simpler) proposal pdf $q(\boldsymbol{\theta})$, and then compute normalized importance weights (IWs) $w^{(i)}$ as

$$w^{(i)*} \propto \frac{\pi(\boldsymbol{\theta}^{(i)})}{q(\boldsymbol{\theta}^{(i)})}, \quad w^{(i)} = \frac{w^{(i)*}}{\sum_{j=1}^{M} w^{(j)*}}, \quad i = 1, \ldots, M.$$

Using Θ^M and the associated weights, we can construct an approximation to the target distribution by means of the discrete random measure

$$\pi^M(d\boldsymbol{\theta}) = \sum_{i=1}^{M} w^{(i)} \delta_{\boldsymbol{\theta}^{(i)}}(d\boldsymbol{\theta}),$$

and approximate (f, π) by the weighted sum

$$(f, \pi^M) = \sum_{i=1}^{M} w^{(i)} f(\boldsymbol{\theta}^{(i)}).$$

Note that the computation of the normalized IWs only requires that both $\pi(\boldsymbol{\theta})$ and $q(\boldsymbol{\theta})$ can be evaluated up to a proportionality constant. Moreover, the same sample generated from $q(\boldsymbol{\theta})$ can be used to approximate integrals of different functions and under different densities $\pi(\boldsymbol{\theta})$.

The proposal density is usually selected so that it is easy to simulate from it. In order to ensure the asymptotic convergence of the approximation (f, π^M), as $M \to \infty$, it is sufficient to select $q(\boldsymbol{\theta})$ such that $q(\boldsymbol{\theta}) > 0$ whenever $\pi(\boldsymbol{\theta}) > 0$ [147]. However, the variance of the estimator is only finite when the importance function has heavier tails than the target pdf.

In order to obtain, if needed, a set of unweighted samples approximately distributed from the target distribution, a so-called *resampling* step can be performed [147, 54]. For simplicity, in this work we only consider multinomial resampling, which consists in drawing samples with replacement from the set $\Theta^M = \{\boldsymbol{\theta}^{(i)}\}_{i=1}^M$ according to the IWs $w^{(i)}$, replicating those samples with high IWs and discarding those with low IWs. The resulting samples $\tilde{\Theta}^M = \{\tilde{\boldsymbol{\theta}}^{(i)}\}_{i=1}^M$ are equally representative of the target distribution and are thus equally weighted, i.e., $\tilde{w}^{(i)} = 1/M$, $i = 1, \ldots, M$, where $\tilde{w}^{(i)}$ denotes the IWs after the resampling step. The resampling step is often used in IS-based techniques to partly alleviate the weight degeneracy problem, eliminating non representative samples. However, some resampling schemes, namely multinomial resampling, increase the variance of the estimators and better alternatives exist. See, e.g., [12, 33], for an overview of resampling techniques.

In Figure 2.2 we illustrate the IS methodology in a simple unidimensional example ($K = 1$). A set of samples $\{\theta^{(i)}\}_{i=1}^M$ is generated from a proposal pdf $q(\theta) = p(\theta)$ and the IWs are computed as the likelihood of those samples $w^{(i)} \propto p(y|\theta^{(i)})$. A resampling step has been performed to produce unweighted samples, and the histogram of the resulting set is shown to fit the target distribution.

When analytically intractable, the likelihood $p(\mathbf{y}|\boldsymbol{\theta})$ given in equation (2.9) can be approximated via IS. Drawing $\{\mathbf{x}^{(i)}\}_{i=1}^M$ from a suitable proposal distribution with pdf $q(\mathbf{x})$ we readily obtain

$$p(\mathbf{y}|\boldsymbol{\theta}) = \int \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x}|\boldsymbol{\theta})}{q(\mathbf{x})} q(\mathbf{x}) d\mathbf{x} \approx \frac{1}{M} \sum_{i=1}^{M} \frac{p(\mathbf{y}|\mathbf{x}^{(i)})p(\mathbf{x}^{(i)}|\boldsymbol{\theta})}{q(\mathbf{x}^{(i)})}.$$

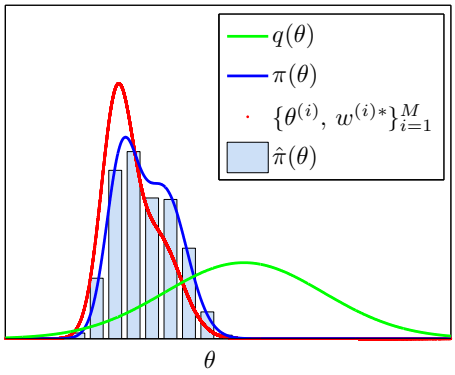

Figure 2.2: Approximation of a target pdf $\pi(\boldsymbol{\theta})$ via IS and resampling.

A frequently used index for the performance of Monte Carlo approximations of probability measures is the effective sample size (ESS) [101, 57]. For IS methods, the ESS may be intuitively interpreted as the relative size of a sample generated from the target distribution with the same variance as the current sample. Even when high values of the ESS do not guarantee a low approximation error, it can be used as an indicator of the numerical stability of the algorithm [57, 55]. The ESS cannot be evaluated exactly but we can compute an approximation based on the set of IWs as [101, 147] $M^{eff} = 1/\sum_{i=1}^{M}(w^{(i)})^2$. The normalized ESS (NESS), is in turn computed as $M^{neff} = M^{eff}/M$ and takes values between 0 and 1. A NESS value close to 1 (low variance of the IWs) suggests a good agreement between the proposal and the target pdf. However, it is generally not enough to establish that the algorithm has converged to the true target, for example, when the target presents multiple modes.

Degeneracy of the IWs

The efficiency of an IS algorithm depends heavily on the choice of the proposal distribution, $q(\boldsymbol{\theta})$. Unless the proposal pdf is well tailored to the target density, the normalized IWs $w^{(i)}$, $i = 1, ..., M$, of a set of samples $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{M}$ present large fluctuations and their maximum, $\max_i w^{(i)}$, is close to one, leading to an extremely low ESS. This situation occurs when the target and the proposal densities are approximately mutually singular, i.e., they (essentially) have disjoint supports [19]. This problem is well known and it is usually termed degeneracy of the weights [101, 57].

The degeneracy of the IWs critically increases with K [19, 21], which has been widely accepted as one of the main drawbacks of IS. However, it can be easily verified (numerically) that IS techniques can suffer from degeneracy even when applied to low dimensional systems. Assume that the target pdf is the posterior pdf $\pi(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y})$ and consider a set of M samples $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{M}$ drawn from the prior pdf $p(\boldsymbol{\theta})$. Assuming conditionally independent observations, the IW associated to the i-th sample is given by

$$w^{(i)} \propto p(\mathbf{y}|\boldsymbol{\theta}^{(i)}) = \prod_{n=1}^{N} p(\mathbf{y}_n|\boldsymbol{\theta}^{(i)}),\ i = 1, \ldots, M. \tag{2.14}$$

Thus, the IWs are obtained from a likelihood consisting of the product of a potentially large number of factors. As the number of observations N increases, the posterior probability concentrates in a smaller region (it becomes sharper), leading to a low probability of obtaining representative samples. This shows how in low dimensional systems degeneracy of the IWs can be motivated by a high number of observations N, unless the computational inference method is explicitly designed to account for this difficulty. For this reason, IS methods have been traditionally avoided for batch estimation due to their inefficiency in complex problems. However, IS has been widely applied as the core of SMC methods.

2.4 Sequential Monte Carlo methods

Sequential Monte Carlo (SMC) methods are a family of simulation-based algorithms that allow to estimate posterior densities of interest in a recursive manner [12, 54]. The use of SMC methods is natural for implementing the Bayesian recursion equations in state-space models [55]. However, the SMC approach actually provides a more general framework which also encompasses inference for static target distributions [42, 47], yielding an alternative to standard batch methods, such as MCMC or PMC. In this section we describe some relevant SMC methods for Bayesian filtering and parameter estimation in static and dynamical models.

2.4.1 SMC for Bayesian filtering: particle filters

Particle filtering is an SMC methodology based on a recursive implementation of the IS technique, also known as sequential IS (SIS). It yields consistent approximations to the optimal Bayesian filter described by equations (2.4) and (2.5), when the hidden state and the observations are

related through a state-space model [10, 51, 73]. Particle filters (PFs) allow to represent nearly any posterior distribution with pdf $\pi(\mathbf{x}_n) = p(\mathbf{x}_n|\mathbf{y}_{1:n}, \boldsymbol{\theta})$ and they are specially suited for difficult nonlinear and/or non-Gaussian problems. PFs provide a very flexible framework which can be applied to complex problems in a vast amount of applications [7, 55, 13]. One specific feature of these techniques that turns out appealing in practice is that they are strictly recursive algorithms. Hence, they keep updating the approximation of the filter distribution as new observations are gathered and the data storage requirements of this algorithm are relatively modest. While often acknowledged as computationally costly, the fast development of powerful computers in recent years has allowed a very wide use of PFs in practice and the appearance of many useful extensions [57, 142, 38, 102].

The selection of the importance function used by a PF significantly affects the performance of the algorithm [57]. In this work we consider the simplest choice of importance function, which is the transition pdf of the state-space model. Thus, at time step n, new samples or *particles* are generated as $\mathbf{x}_n^{(j)} \sim p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(j)}, \boldsymbol{\theta})$, $j = 1, \ldots, J$. This choice of importance function is somewhat inefficient because new state particles are generated ignoring the current observation [57]. However, it yields a very simple expression for the IWs and is adequate for the purposes of this work. With this simple importance function, the resulting unnormalized IWs are of the form $w_n^{(j)*} \propto w_{n-1}^{(j)} p(\mathbf{y}_n|\mathbf{x}_n^{(j)})$. To compute the IWs, the target distribution and the importance function are required to be known only up to a normalizing constant. The sequential implementation of IS suffers from the degeneracy of the IWs or sample impoverishment, which represents a major drawback of this technique [21, 19]. The recursive update of the IWs is bound to fail in the long run, when most of the samples attain negligible IWs and only a few contribute to the approximation of the target distribution [57, 55].

The solution to mitigate the degeneracy problem proposed in [73] is to allow rejuvenation of the set of samples by probabilistically replicating samples with high IWs and removing samples with low IWs, by means of a resampling step. When the size of the observation vector $\mathbf{y}_n$ is relatively small and the samples $\mathbf{x}_n$, even sampled from the prior distribution $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \boldsymbol{\theta})$, are sufficiently close to the observations, degeneracy becomes only slight and can be indeed mitigated by a simple resampling step. If resampling is performed at each sequential step, the resulting scheme is the classical bootstrap filter [73, 57]. In this case, the IWs do not depend on the past trajectory of the particles but only on the likelihood of the current

samples, i.e., $w_n^{(j)} \propto p(\mathbf{y}_n|\mathbf{x}_n^{(j)})$. The standard PF with a prior transition kernel and multinomial resampling is outlined in Table 2.1.

Table 2.1: Standard particle filter with prior transition kernel [51, 73].

Initialization ($n = 0$):

1. Draw a collection of J samples $\{\mathbf{x}_0^{(j)}\}_{j=1}^J \sim p(\mathbf{x}_0)$ from the prior distribution of the state $p(\mathbf{x}_0)$.

Recursive step ($n = 1, \ldots, N$):

1. Draw $\{\mathbf{x}_n^{(j)}\}_{j=1}^J \sim p(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^{(j)}, \boldsymbol{\theta})$ from the transition distribution and set $\mathbf{x}_{0:n}^{(j)} = [\tilde{\mathbf{x}}_{0:n-1}^{(j)\top}, \mathbf{x}_n^{(j)\top}]^\top$.
2. Compute normalized IWs $\omega_n^{(j)} \propto p(\mathbf{y}_n|\mathbf{x}_n^{(j)})$, $j = 1, \ldots, J$.
3. Resample J times with replacement from $\{\mathbf{x}_{0:n}^{(j)}\}_{j=1}^J$ according to the weights $\{\omega_n^{(j)}\}_{j=1}^J$, yielding a set of equally weighted samples $\{\tilde{\mathbf{x}}_{0:n}^{(j)}, \tilde{\omega}_n^{(j)}\}_{j=1}^J$, with $\tilde{\omega}_n^{(j)} = 1/J$.

At each time step, an approximation of the filtering density $\pi(\mathbf{x}_n) = p(\mathbf{x}_n|\mathbf{y}_{1:n}, \boldsymbol{\theta})$ can be obtained as the following discrete random measure

$$\hat{\pi}^J(d\mathbf{x}_n) = \sum_{j=1}^J \omega_n^{(j)} \delta_{\mathbf{x}_n^{(j)}}(d\mathbf{x}_n), \tag{2.15}$$

and integrals w.r.t. $\pi(\mathbf{x}_n)d\mathbf{x}_n$ can be approximated as

$$(f, \pi) \approx (f, \hat{\pi}^J) = \sum_{j=1}^J \omega_n^{(j)} f(\mathbf{x}_n^{(j)}),$$

where f is some integrable function of $\mathbf{x}_n$. In particular, an approximation of the posterior mean of $\mathbf{x}_n$ can be obtained as

$$\hat{\mathbf{x}}_n^J = \sum_{j=1}^J \omega_n^{(j)} \mathbf{x}_n^{(j)}, \tag{2.16}$$

which corresponds to the MMSE estimate of the hidden state. Notice that estimation should be carried out using the weighted particles, since the particle representation after resampling has a higher Monte Carlo error [51].

Figure 2.3 illustrates the performance of the PF in a typical practical application, namely target tracking in a wireless sensor network based on received signal strength (RSS) measurements [1]. The hidden state of the target consists of its position and velocity, which evolve according to a transition model. A network of sensors placed at known locations collect the RSS observations at each time instant (*left* plot). The PF allows to estimate the hidden state of the target based on sequential observations in a purely recursive manner.

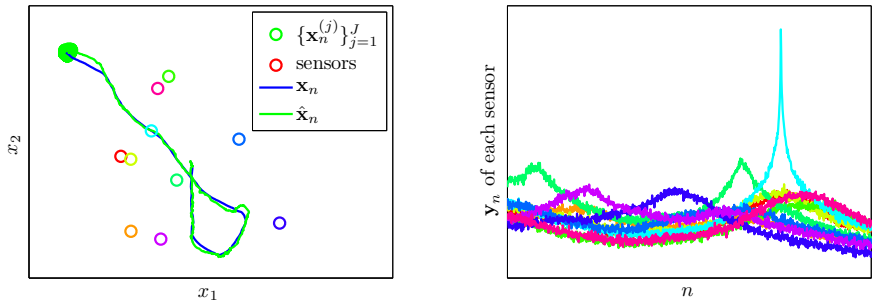

Figure 2.3: Example of the performance of the PF applied to a target tracking problem in a wireless sensor network based on RSS measurements.

Approximation of the likelihood $p(\mathbf{y}|\boldsymbol{\theta})$ via particle filtering

The standard PF described here allows to obtain an approximation of the likelihood of the parameters $p(\mathbf{y}|\boldsymbol{\theta})$ in equation (2.9) using the decomposition

$$p(\mathbf{y}|\boldsymbol{\theta}) = p(\mathbf{y}_1|\boldsymbol{\theta}) \prod_{n=2}^{N} p(\mathbf{y}_n|\mathbf{y}_{1:n-1}, \boldsymbol{\theta}), \tag{2.17}$$

where the individual predictive likelihood terms are integrals w.r.t. the predictive density of the state $p(\mathbf{x}_n|\mathbf{y}_{1:n-1}, \boldsymbol{\theta})$, namely,

$$p(\mathbf{y}_n|\mathbf{y}_{1:n-1}, \boldsymbol{\theta}) = \int p(\mathbf{y}_n|\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{y}_{1:n-1}, \boldsymbol{\theta}) d\mathbf{x}_n = E_{p(\mathbf{x}_n|\mathbf{y}_{1:n-1}, \boldsymbol{\theta})}[p(\mathbf{y}_n|\mathbf{x}_n)].$$

The predictive distribution of the state can be approximated via a PF as

$$p(\mathbf{x}_n|\mathbf{y}_{1:n-1},\boldsymbol{\theta})d\mathbf{x}_n \approx \sum_{j=1}^{J} \omega_{n-1}^{(j)} \delta_{\mathbf{x}_n^{(j)}}(d\mathbf{x}_n), \tag{2.18}$$

based on the previous set of normalized IWs $\omega_{n-1}^{(j)}$ and a set of new samples $\mathbf{x}_n^{(j)}$ drawn from the transition kernel $p(\mathbf{x}_n|\mathbf{x}_{n-1}^{(j)},\boldsymbol{\theta})$, as in the bootstrap filter. The predictive likelihood approximation is, in turn, given by [31]

$$p(\mathbf{y}_n|\mathbf{y}_{1:n-1},\boldsymbol{\theta}) \approx \frac{1}{J}\sum_{j=1}^{J} p(\mathbf{y}_n|\mathbf{x}_n^{(j)}) = \frac{1}{J}\sum_{j=1}^{J} \omega_n^{(j)*}. \tag{2.19}$$

This recursive likelihood approximation is much more efficient that the ones based on exact Monte Carlo or IS, and is widely used in practice [31, 6]. This technique only requires a low number of particles J [31].

2.4.2 SMC for parameter estimation

SMC techniques are quite obviously related to the Bayesian filtering problem in state-space models, hence the success of PFs. However, from a more general point of view, SMC methods can be used for other inference tasks. In particular, SMC algorithms can also be constructed to approximate the posterior pdf $p(\boldsymbol{\theta}|\mathbf{y})$ of a vector of random parameters, in high-dimensional batch problems of the type described in Section 2.2.1. In this section we review some of the SMC methods proposed for estimation of parameters in non-sequential setups.

Iterated batch IS algorithm

In [41] an SMC method is proposed for the approximation of a static posterior distribution with density $\pi(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y})$, termed iterated batch IS (IBIS). This method explores the sequence of partial posterior distributions of the form

$$\pi_\ell(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y}_{1:n_\ell}), \quad \ell = 1,\ldots,L, \quad 0 \leq n_\ell \leq N, \tag{2.20}$$

incorporating a set of new observations at each step and recursively updating the IWs. The total number of observations considered up to step ℓ is denoted by n_ℓ and satisfies $n_1 = 0$ and $n_L = N$ (hence $\pi_1(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$ and $\pi_L(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta})$). When considering large data sets, this approach may result

in a reduced computational complexity compared to standard methods such as MCMC [41]. Additionally, it is claimed to alleviate numerical problems arising when the likelihood function is extremely peaky, by including the observations in a sequential manner, which results in a beneficial tempering effect [41, 47]. The IBIS algorithm is outlined in Table 2.2.

Table 2.2: Iterated batch IS algorithm [41].

Initialization ($\ell = 1$):

1. Draw a collection of samples $\tilde{\boldsymbol{\theta}}_1^{(i)} \sim p(\boldsymbol{\theta})$, $i = 1, \ldots, M$.

Sequential step ($\ell = 2, \ldots, L$):

1. Draw $\boldsymbol{\theta}_\ell^{(i)} \sim q_\ell(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}_{\ell-1}^{(i)})$, for $i = 1, \ldots, M$, where $q_\ell(\boldsymbol{\theta})$ is a transition kernel with stationary distribution $\pi_\ell(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y}_{1:n_\ell})$.
2. Compute normalized IWs as
$$w_\ell^{(i)} \propto p(\mathbf{y}_{n_{\ell-1}+1:n_\ell}|\mathbf{y}_{1:n_{\ell-1}}, \boldsymbol{\theta}_\ell^{(i)}), \quad i = 1, \ldots, M.$$
3. Resample $\{\boldsymbol{\theta}_\ell^{(i)}, w_\ell^{(i)}\}_{i=1}^M$ to obtain unweighted samples $\{\tilde{\boldsymbol{\theta}}_\ell^{(i)}\}_{i=1}^M$.

This method is suited for problems where the observations are either independent or Markov, in which case the weights can be more easily computed. When the observations are conditionally independent given $\boldsymbol{\theta}$ the IWs reduce to

$$w_\ell^{(i)} \propto p(\mathbf{y}_{n_{\ell-1}+1:n_\ell}|\mathbf{y}_{1:n_{\ell-1}}, \boldsymbol{\theta}_\ell^{(i)}) = \prod_{l=1}^{n_\ell - n_{\ell-1}} p(\mathbf{y}_{n_{\ell-1}+l}|\boldsymbol{\theta}_\ell^{(i)}).$$

Note, however, that this method strictly requires that the likelihood function $p(\mathbf{y}|\boldsymbol{\theta})$ can be evaluated. In general nonlinear, non-Gaussian state-space models the evaluation of the likelihood requires integrating out the latent states $\mathbf{x}$, which constraints the application of the IBIS algorithm in such settings [42]. Note that the IBIS algorithm is sequential but not recursive, since for each sample of $\boldsymbol{\theta}$ the computation of the weights requires considering all the observations.

The efficiency of this scheme can be highly sensitive to the selection of the transition kernel $q_\ell(\boldsymbol{\theta})$. The authors of [41] propose to use a model-

independent Gaussian kernel, with moments matched to the latest sample approximation of the posterior. The sample incorporation schedule is also a delicate issue. The suggestion in [41] is to incorporate the observations one after the other while checking the ESS, until it drops below a certain threshold. This mechanism can yield a geometric rate for the incorporation of the data points [41]. The algorithm is also sensitive to the ordering of the observations and may degenerate strongly in the first iterations [41].

Annealed IS

In [130] a method termed annealed IS (AIS) is proposed, whose goal is also the approximation of a static target distribution $\pi(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y})$. The AIS algorithm moves from a tractable distribution $\pi_1(\boldsymbol{\theta})$, which can be selected as the prior $p(\boldsymbol{\theta})$ in the Bayesian framework, to the target pdf $\pi_L(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta})$, via a sequence of artificial (tempered) densities constructed as

$$\pi_\ell(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta})^{\beta_\ell} p(\boldsymbol{\theta})^{1-\beta_\ell}, \quad \ell = 1, \ldots, L, \tag{2.21}$$

where the exponents β_ℓ are selected such that $0 = \beta_1 < \ldots < \beta_L = 1$. These auxiliary target distributions must satisfy that $\pi_\ell(\boldsymbol{\theta}) > 0$ wherever $\pi_{\ell+1}(\boldsymbol{\theta}) > 0$ (the support decreases) and it must be possible to evaluate π_ℓ up to a normalizing constant. Additionally, for each $\ell = 2, \ldots, L$, we need to be able to simulate from a Markov transition kernel $q_\ell(\boldsymbol{\theta}|\boldsymbol{\theta}_{\ell-1}^{(i)})$, that leaves π_ℓ invariant. The transition kernels can be constructed in any of the usual ways (e.g., using MCMC techniques). The AIS algorithm generates a set of M weighted samples $\{\boldsymbol{\theta}_L^{(i)}, w^{(i)}\}_{i=1}^M$, which allow to approximate integrals of the form (2.10). The AIS algorithm is outlined in Table 2.3.

Table 2.3: Annealed IS algorithm [130].

For each sample $i = 1, \ldots, M$:

1. Generate a sequence of points $\{\boldsymbol{\theta}_1^{(i)}, \ldots, \boldsymbol{\theta}_L^{(i)}\}$ as follows: $\boldsymbol{\theta}_1^{(i)} \sim \pi_1(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$, $\boldsymbol{\theta}_2^{(i)} \sim q_2(\boldsymbol{\theta}|\boldsymbol{\theta}_1^{(i)})$, ..., $\boldsymbol{\theta}_L^{(i)} \sim q_L(\boldsymbol{\theta}|\boldsymbol{\theta}_{L-1}^{(i)})$.
2. Compute the IWs associated to the path as

$$w^{(i)} = \frac{\pi_2(\boldsymbol{\theta}_1^{(i)})}{\pi_1(\boldsymbol{\theta}_1^{(i)})} \cdots \frac{\pi_L(\boldsymbol{\theta}_L^{(i)})}{\pi_{L-1}(\boldsymbol{\theta}_L^{(i)})}.$$

SMC samplers

The SMC sampler of [47] is a methodology to approximate a sequence of probability distributions with densities $\pi_\ell(\boldsymbol{\theta})$, $\ell = 1, \ldots, L$, which can be applied in the batch setting described in Section 2.2.1 for the approximation of a fixed posterior pdf $\pi(\boldsymbol{\theta}) = \pi_L(\boldsymbol{\theta})$.

The target densities $\pi_\ell(\boldsymbol{\theta})$, $\ell = 1, \ldots, L$, are constructed in such a way that consecutive π_ℓ's do not differ significantly, so that samples can be moved from one to the other in a sensible way using Markov kernels. This method provides a general framework which includes as particular cases some important batch algorithms considered in this work, such as IBIS [41], AIS [130] or PMC [32]. Other fields of applicability of the SMC sampler include global optimization, choosing $\pi_\ell(\boldsymbol{\theta}) \propto \pi(\boldsymbol{\theta})^{\phi_\ell}$ with increasing ϕ_ℓ, or estimation of the probability of a rare event [47].

A general SMC sampler relies on the construction of a sequence of artificial joint densities of increasing dimension, namely

$$\alpha_\ell(\boldsymbol{\theta}_{1:\ell}) = \pi_\ell(\boldsymbol{\theta}_\ell) \prod_{r=1}^{\ell-1} b_r(\boldsymbol{\theta}_r|\boldsymbol{\theta}_{r+1}), \tag{2.22}$$

where $b_r(\boldsymbol{\theta}_r|\boldsymbol{\theta}_{r+1})$ is the density of an arbitrary *backward* kernel. By construction, the joint pdf in (2.22) has $\pi_\ell(\boldsymbol{\theta}_\ell)$ as a marginal, i.e.,

$$\int \ldots \int \alpha_\ell(\boldsymbol{\theta}_{1:\ell}) d\boldsymbol{\theta}_{\ell-1} \ldots d\boldsymbol{\theta}_1 = \pi_\ell(\boldsymbol{\theta}_\ell). \tag{2.23}$$

If we choose a sequence of *forward* kernels with densities $q_1(\boldsymbol{\theta})$, $q_\ell(\boldsymbol{\theta}|\boldsymbol{\theta}_{\ell-1})$, $\ell = 2, \ldots, L$, it is possible to run a standard SIS algorithm [57] to approximate the measure $\alpha_\ell(\boldsymbol{\theta}_{1:\ell}) d\boldsymbol{\theta}_{1:\ell}$ (and its marginals). The resulting procedure is shown in Table 2.4. In [47] resampling is performed only when the ESS falls below some threshold, but here we assume that resampling is performed at every sequential step.

The performance of SMC samplers is highly dependent on the selection of the target distributions π_ℓ and the transition kernels q_ℓ that are used to explore the space of $\boldsymbol{\theta}$. The sequence of transition kernels can be constructed, e.g., as independent proposals choosing $q_\ell(\boldsymbol{\theta}|\boldsymbol{\theta}_{\ell-1}) = q_\ell(\boldsymbol{\theta})$, local random-walk moves with a standard smoothing kernel, MCMC moves with invariant distribution π_ℓ, or approximate Gibbs moves [47].

A technique related to the SMC sampler and the AIS algorithm is analyzed in [20], where a sequence of artificial targets is constructed as flattened versions of the target pdf, i.e., $\pi_\ell(\boldsymbol{\theta}) \propto \pi(\boldsymbol{\theta})^{\beta_\ell}$, where $0 < \beta_1 <$

Table 2.4: Sequential Monte Carlo sampler [47].

Initialization ($\ell = 1$):

1. Draw $\boldsymbol{\theta}_1^{(i)}$ from $q_1(\boldsymbol{\theta}_1)$, $i = 1, \ldots, M$.
2. Set the initial (normalized) IWs as $w_1^{(i)} \propto \pi_1(\boldsymbol{\theta}_1^{(i)})/q_1(\boldsymbol{\theta}_1^{(i)})$.
3. Resample to obtain an unweighted set $\{\tilde{\boldsymbol{\theta}}_1^{(i)}\}_{i=1}^M$.

Iteration ($\ell = 2, \ldots, L$):

1. Draw $\boldsymbol{\theta}_\ell^{(i)}$ from $q_\ell(\boldsymbol{\theta}_\ell|\tilde{\boldsymbol{\theta}}_{\ell-1}^{(i)})$, $i = 1, ..., M$.
2. Compute (normalized) IWs

$$w_\ell^{(i)} \propto \frac{\pi_\ell(\boldsymbol{\theta}_\ell^{(i)}) b_{\ell-1}(\tilde{\boldsymbol{\theta}}_{\ell-1}^{(i)}|\boldsymbol{\theta}_\ell^{(i)})}{\pi_{\ell-1}(\tilde{\boldsymbol{\theta}}_{\ell-1}^{(i)}) q_\ell(\boldsymbol{\theta}_\ell^{(i)}|\tilde{\boldsymbol{\theta}}_{\ell-1}^{(i)})}. \tag{2.24}$$

3. Resample according to the IWs $w_\ell^{(i)}$ to obtain the set $\{\tilde{\boldsymbol{\theta}}_\ell^{(i)}\}_{i=1}^M$.

$\ldots < \beta_L = 1$. The authors of [20] provide a convergence analysis of the SMC technique when the dimension of the parameter space becomes large.

2.4.3 Bayesian filtering with parameter estimation

In this section we address the problem of approximating the sequence of posterior densities $p(\boldsymbol{\theta}, \mathbf{x}_{0:n}|\mathbf{y}_{1:n})$, $n \geq 1$, in a sequential manner as observations $\mathbf{y}_n$ become available [88, 89] (the filtering density $p(\boldsymbol{\theta}, \mathbf{x}_n|\mathbf{y}_{1:n})$ can be obtained as a marginal of the full posterior pdf). This is a challenging problem, given that the dimension of the target distribution increases over time and the observation sequence $\mathbf{y}_{1:n}$ is non-ergodic, as it is associated to a single realization of the random parameter $\boldsymbol{\theta}$ [139, 42, 44].

To estimate the posterior density $p(\boldsymbol{\theta}, \mathbf{x}_{0:n}|\mathbf{y}_{1:n})$ it may seem natural to apply a standard PF considering the unknown parameter vector $\boldsymbol{\theta}$ as a component of the state with no dynamics. Unfortunately, this solution does not allow for the exploration of the parameter space and thus the PF would yield a poor performance [7]. A possible solution for this problem

requires the introduction of artificial dynamics for the fixed parameters, such that $\boldsymbol{\theta}$ is "moved around" using a noise component with a small variance [93, 113]. This transforms the static parameter $\boldsymbol{\theta}$ into a slowly varying dynamic one. However, this technique requires a significant amount of tuning and introduces a bias which is hard to quantify [7].

Also ML estimation of the parameters has been considered in the literature, which is obtained via gradient optimization methods [109], or via expectation maximization [7, 88]. However, no convergence results have been provided for these techniques for general state-space models.

Another possibility is to add MCMC steps (typically a random walk) to induce diversity among the particles, with the joint posterior $p(\boldsymbol{\theta}, \mathbf{x}_{0:n}|\mathbf{y}_{1:n})$ as stationary pdf [4, 67, 152]. This is a neat solution, since the model is not artificially altered, and it can perform well in low dimensional problems. However, this approach also suffers from the degeneracy problem [8, 88].

In [139] a method was proposed where samples $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{M}$ are generated from the prior distribution and a PF for each sample is run to approximate the posterior pdf $p(\mathbf{x}_{0:n}|\mathbf{y}_{1:n}, \boldsymbol{\theta}^{(i)})$. The algorithm is recursive, but it can require a very large number of samples in the parameter space.

SMC^2 algorithm

The recently proposed SMC square (SMC^2) algorithm allows for the sequential estimation of parameters and hidden states in a general state-space model, recursively exploring the sequence of posterior pdf $p(\boldsymbol{\theta}, \mathbf{x}_{0:n}|\mathbf{y}_{1:n})$ [42]. SMC^2 can be seen as a combination of IBIS and PF, where we attach a PF to each sample of $\boldsymbol{\theta}$ to compute unbiased estimates of the marginal likelihood $p(\mathbf{y}_n|\mathbf{y}_{1:n-1}, \boldsymbol{\theta})$, in cases when it is not available in closed form. SMC^2 displays a structure of nested PFs that has inspired its name. The same as the IBIS method of [41], the SMC^2 algorithm requires to run a PF from scratch up to time n for each new sample of the parameters. Thus, it is a sequential but offline algorithm, whose computational load increases, at least, with the square of the length of the observation sequence [42]. This method is mainly suited for sequential setups but it may result computationally advantageous also in batch estimation scenarios. Opposite to standard batch methods such as MCMC or PMC, which usually have to be rerun for each time horizon, the SMC^2 algorithm allows for a partial reuse of the computations as new observations are collected. The SMC^2 algorithm is outlined in Table 2.5.

An approximation of the posterior pdf of the parameters $p(\boldsymbol{\theta}|\mathbf{y}_{1:n})$ can

Table 2.5: SMC^2 algorithm [42].

Initialization ($n = 1$):

1. Draw $\tilde{\boldsymbol{\theta}}_1^{(i)} \sim p(\boldsymbol{\theta})$, $i = 1, \ldots, M$.

Sequential step ($n = 2, \ldots, N$):

1. Draw $\boldsymbol{\theta}_n^{(i)} \sim q_n(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}_{n-1}^{(i)})$, $i = 1, \ldots, M$, from a suitable kernel q_n.
2. For $i = 1, \ldots, M$ run a PF with J particles targeting $p(\mathbf{x}_n|\mathbf{y}_{1:n}, \boldsymbol{\theta}_n^{(i)})$ and compute the IWs for $\boldsymbol{\theta}_n^{(i)}$ as in equation (2.19), i.e.,

$$w_n^{(i)} \propto p(\mathbf{y}_n|\mathbf{y}_{1:n-1}, \boldsymbol{\theta}_n^{(i)}) \approx \frac{1}{J}\sum_{j=1}^{J} p(\mathbf{y}_n|\mathbf{x}_n^{(i,j)}), \quad i = 1, \ldots, M,$$

   where $\mathbf{x}_n^{(i,j)}$ denotes the j-th state particle of the PF for $\boldsymbol{\theta}_n^{(i)}$. Compute an approximation $\hat{\mathbf{x}}_n^{(i)J}$ of the posterior mean of $\mathbf{x}_n$ given $\boldsymbol{\theta}_n^{(i)}$ and $\mathbf{y}_n$, as in equation (2.16).
3. Resample the weighted set $\{\boldsymbol{\theta}_n^{(i)}, w_n^{(i)}\}_{i=1}^M$ to yield unweighted samples $\{\tilde{\boldsymbol{\theta}}_n^{(i)}\}_{i=1}^M$.

be constructed at each time step $n = 1, \ldots, N$ as

$$\hat{p}^{M,J}(d\boldsymbol{\theta}|\mathbf{y}_{1:n}) = \sum_{i=1}^{M} w_n^{(i)} \delta_{\boldsymbol{\theta}_n^{(i)}}(d\boldsymbol{\theta})$$

and the posterior pdf of the hidden state $p(\mathbf{x}_n|\mathbf{y}_{1:n})$ can be approximated, in turn, as

$$\hat{p}^{M,J}(d\mathbf{x}_n|\mathbf{y}_{1:n}) = \sum_{i=1}^{M} w_n^{(i)} \delta_{\hat{\mathbf{x}}_n^{(i)J}}(d\mathbf{x}_n).$$

A truly online version of the SMC^2 algorithm, with all computations being strictly recursive, has been recently proposed in [44]. Similarly to SMC^2, the algorithm consists of two layers of nested PFs. However, this algorithm is recursive and admits an online implementation with a constant computational cost per time step. The convergence of this online version

of the algorithm, however, is subject to the family of conditional filters $p(\mathbf{x}_n|\mathbf{y}_{1:n}, \boldsymbol{\theta})$ being continuous w.r.t. the parameter $\boldsymbol{\theta}$. See [139] for a discussion on the continuity of the filter distribution.

2.5 Markov chain Monte Carlo methods

Markov chain Monte Carlo (MCMC) methods are a family of algorithms that allow to obtain samples approximately distributed from a possibly multidimensional static target distribution $\pi(\boldsymbol{\theta})$ and, consequently, to approximate integrals of the form (f, π) [147, 66].

MCMC methods have been traditionally favoured by statisticians to sample from complex distributions. The basic principle behind MCMC methods is the following: for an arbitrary initial point $\boldsymbol{\theta}^{(1)}$, an irreducible Markov chain $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{I}$ is generated using a transition kernel $q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(i-1)})$ with stationary distribution $\pi(\boldsymbol{\theta})$. The Markov property guarantees that the state $\boldsymbol{\theta}^{(i)}$ only depends on the previous state $\boldsymbol{\theta}^{(i-1)}$. A Markov chain is said to have a stationary or equilibrium probability distribution $\pi(\boldsymbol{\theta})$ when it satisfies that $\boldsymbol{\theta}^{(i)} \sim \pi(\boldsymbol{\theta})$ if $\boldsymbol{\theta}^{(i-1)} \sim \pi(\boldsymbol{\theta})$. The irreducibility property measures the sensitivity of the Markov chain to the initial conditions and states that from each state it is possible to reach every other state [147].

The initial samples of the chain may not accurately represent the desired distribution, and should be discarded. This is often referred to as the "burn in" period of the chain and it accounts for the number of iterations that the chain is expected to take in order to reach its stationary distribution [66]. The sample $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{I}$ generated by an MCMC algorithm presents correlations, which slows down convergence of the approximation $\frac{1}{I}\sum_{i=1}^{I} f(\boldsymbol{\theta}^{(i)})$ toward (f, π). If reducing the correlation among samples is required, we can thin the resulting chain by only taking every j-th value. The selection of the number of iterations I and the assessment of convergence of the chain to the equilibrium distribution $\pi(\boldsymbol{\theta})$ are also delicate issues.

A huge number of methods have been proposed based on the MCMC principle. One of the most popular algorithms is the Gibbs sampler [64, 34], which allows to sample from multidimensional target densities $\pi(\boldsymbol{\theta}) = \pi(\theta_1, \ldots, \theta_K)$ by sequentially sampling from univariate full conditional distributions $\pi(\theta_k|\boldsymbol{\theta}_{\setminus k})$, where $\boldsymbol{\theta}_{\setminus k} = [\theta_1, \ldots, \theta_{k-1}, \theta_{k+1}, \theta_K]^\top$ denotes the vector containing all parameters in $\boldsymbol{\theta}$ except for θ_k. This method is applicable when the joint target distribution is intractable, but the conditional distribution of each variable is easy to sample from. It is a rather restrictive approach since it requires the knowledge of these conditional

distributions [147].

In this section we describe the very general Metropolis-Hastings algorithm [125, 77, 147], which imposes minimal requirements on the target density and allows for a wide choice of possible implementations. We also review the particle MCMC method [6], which provides a powerful tool for sampling from posterior distributions $p(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$ and their marginals, based on approximations of the likelihood function $p(\mathbf{y}|\boldsymbol{\theta})$ computed by means of a PF. Many other relevant MCMC methods exist which are out of the scope of this work [131, 116, 75, 58, 68].

2.5.1 Metropolis-Hastings algorithm

The Metropolis-Hastings (MH) algorithm [125, 77] allows to samples from any posterior target pdf $\pi(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y})$ which can be evaluated up to a normalizing constant. It requires the definition of an initial proposal distribution, which can be selected as the prior in the Bayesian framework, that is used to generate the starting point of the Markov chain $\boldsymbol{\theta}^{(1)}$. It also requires the selection of a transition kernel $q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(i-1)})$ (a proposal distribution), which is easy to simulate from and is either symmetric ($q(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)}) = q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^{(i)})$) or can be evaluated up to a normalizing constant independent of $\boldsymbol{\theta}$. The transition kernel allows to explore the space of $\boldsymbol{\theta}$, probabilistically accepting those samples that are located in the high probability region of the target distribution. The support of the proposal distribution must contain that of the target, to guarantee convergence. Thus, the MH algorithm constructs a Markov chain $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{I}$, which can be used to approximate integrals (f, π). The MH algorithm for a Bayesian approach is shown in Table 2.6.

This algorithm always accepts samples $\boldsymbol{\theta}^{\star}$ such that the ratio $p(\boldsymbol{\theta}^{\star}|\mathbf{y})/q(\boldsymbol{\theta}^{\star}|\boldsymbol{\theta}^{(i-1)})$ is increased w.r.t. the previous sample. Additionally, it may accept samples $\boldsymbol{\theta}^{\star}$ such that this ratio is decreased. It is very common to use a Gaussian random walk proposal pdf $q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(i-1)}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(i-1)}, \sigma^2\mathbf{I})$ centered on the previous value of the chain and with covariance matrix $\sigma^2\mathbf{I}$. In this case, the acceptance probability for the i-th element of the chain reduces to $\min\left\{1, \frac{p(\boldsymbol{\theta}^{\star}|\mathbf{y})}{p(\boldsymbol{\theta}^{(i-1)}|\mathbf{y})}\right\}$, that is, it only depends on the target pdf. However, the random walk MH algorithm is known to fail in large-dimensional and disconnected supports, because it can take too long to explore the space of interest [147, 66]. The selection of the tunning parameter σ^2 heavily determines the performance of the algorithm. If σ^2 is too large, the acceptance probability drops dramatically. On the contrary, if σ^2 is low, almost all samples are accepted but the exploration of the

Table 2.6: Metropolis-Hastings algorithm targeting $p(\boldsymbol{\theta}|\mathbf{y})$ [77].

Initialization $(i = 1)$**:**

1. Draw the starting point from the prior distribution $\boldsymbol{\theta}^{(1)} \sim p(\boldsymbol{\theta})$.

Iteration $(i = 2, \ldots, I)$**:**

1. Draw a proposed sample $\boldsymbol{\theta}^{\star} \sim q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(i-1)})$.
2. With probability

$$\min\left\{1, \frac{p(\mathbf{y}|\boldsymbol{\theta}^{\star})p(\boldsymbol{\theta}^{\star})}{p(\mathbf{y}|\boldsymbol{\theta}^{(i-1)})p(\boldsymbol{\theta}^{(i-1)})} \times \frac{q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^{\star})}{q(\boldsymbol{\theta}^{\star}|\boldsymbol{\theta}^{(i-1)})}\right\}$$

accept the move setting $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{\star}$. Otherwise store the current value $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$.

space is poor and the chain is likely to get stuck in local modes or in low-probability regions. In both cases, the resulting Markov chain turns out highly correlated, i.e., the chain presents poor mixing properties [147].

In Figure 2.4 we show two examples of the performance of the MH algorithm. In the *left* plot we consider a univariate target pdf, which is approximated based on a Markov chain starting at a random point. In the *right* plot, the target pdf is bidimensional, and it can be observed how the Markov chain reaches the region of high probability after a few iterations.

2.5.2 Particle MCMC

Even though the convergence of MCMC algorithms is guaranteed under mild assumptions, they often present poor performance in practice, specially in high-dimensional problems, when the proposal distributions cannot be properly chosen. The particle MCMC (PMCMC) method [6] relies on a combination of MCMC and SMC methods, which takes advantage of the strengths of both components. In the PMCMC framework, SMC algorithms are used to design efficient proposal distributions for MCMC algorithms, specially for inference in state-space models.

The particle marginal Metropolis-Hastings (PMMH) algorithm is a PMCMC method, originally proposed in [6] for Monte Carlo sampling from

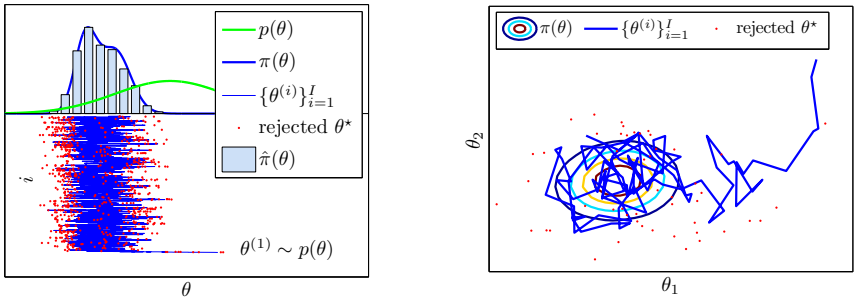

Figure 2.4: Performance of the MH algorithm in a unidimensional (*left*) and a bidimensional (*right*) example.

the full posterior pdf $p(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$, jointly updating $\boldsymbol{\theta}$ and $\mathbf{x}$. The PMMH scheme suggests a proposal mechanism of the form $q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(i-1)})\hat{p}^J(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$. A new candidate in the parameter space, $\boldsymbol{\theta}^\star$, is drawn from an arbitrary proposal distribution $q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(i-1)})$, while the new candidate in the variable space, $\mathbf{x}^\star$, is generated using an approximation of the posterior marginal $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}^\star)$ constructed by means of a PF with J particles and denoted $\hat{p}^J(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}^\star)$.

The probability of accepting the proposed pair $(\boldsymbol{\theta}^\star, \mathbf{x}^\star)$ is computed using the unbiased estimate $\hat{p}^J(\mathbf{y}|\boldsymbol{\theta}^\star)$ of the marginal likelihood of $\boldsymbol{\theta}^\star$, computed, again, by way of a PF with J particles as in equations (2.17) and (2.19). The main feature of the PMMH algorithm is that the invariant distribution of the generated Markov chain is the posterior distribution of interest, $p(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$, independently of the number of samples J in the PF. However, a large value of J yields better mixing properties. The PMMH algorithm is reproduced in Table 2.7.

After removing the initial burn-in samples and thinning the output, we obtain a Markov chain $\{\boldsymbol{\theta}^{(i)}, \mathbf{x}^{(i)}\}_{i=1}^{M}$ with M correlated samples. Then, we may construct a sample approximation of the marginal posterior distributions of the parameters $\boldsymbol{\theta}$ and the populations $\mathbf{x}$, as

$$\hat{p}^{M,J}(d\boldsymbol{\theta}|\mathbf{y}) = \frac{1}{M}\sum_{i=1}^{M}\delta_{\boldsymbol{\theta}^{(i)}}(d\boldsymbol{\theta}) \quad \text{and} \quad \hat{p}^{M,J}(d\mathbf{x}|\mathbf{y}) = \frac{1}{M}\sum_{i=1}^{M}\delta_{\mathbf{x}^{(i)}}(d\mathbf{x}),$$

respectively. The approximation of the full joint posterior is of the form

$$\hat{p}^{M,J}(d\boldsymbol{\theta}, d\mathbf{x}|\mathbf{y}) = \frac{1}{M}\sum_{i=1}^{M}\delta_{(\boldsymbol{\theta}^{(i)},\mathbf{x}^{(i)})}(d\boldsymbol{\theta}, d\mathbf{x}).$$

Table 2.7: Particle MCMC algorithm targeting $p(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$ [6].

Initialization ($i = 1$):

1. Sample $\boldsymbol{\theta}^{(1)} \sim p(\boldsymbol{\theta})$.
2. Run a PF targeting $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}^{(1)})$. Draw $\mathbf{x}^{(1)} \sim \hat{p}^J(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}^{(1)})$ from the PF posterior approximation and compute the marginal likelihood estimate $\hat{p}^J(\mathbf{y}|\boldsymbol{\theta}^{(1)})$.

Iteration ($i = 2, \ldots, I$):

1. Sample $\boldsymbol{\theta}^\star \sim q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(i-1)})$.
2. Run a PF targeting $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}^\star)$. Draw $\mathbf{x}^\star \sim \hat{p}^J(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}^\star)$ and compute $\hat{p}^J(\mathbf{y}|\boldsymbol{\theta}^\star)$.
3. With probability

$$\min\left\{1, \frac{\hat{p}^J(\mathbf{y}|\boldsymbol{\theta}^\star)p(\boldsymbol{\theta}^\star)}{\hat{p}^J(\mathbf{y}|\boldsymbol{\theta}^{(i-1)})p(\boldsymbol{\theta}^{(i-1)})} \times \frac{q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^\star)}{q(\boldsymbol{\theta}^\star|\boldsymbol{\theta}^{(i-1)})}\right\}$$

accept the move setting $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^\star$, $\mathbf{x}^{(i)} = \mathbf{x}^\star$ and $\hat{p}^J(\mathbf{y}|\boldsymbol{\theta}^{(i)}) = \hat{p}^J(\mathbf{y}|\boldsymbol{\theta}^\star)$. Otherwise store the current values $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$, $\mathbf{x}^{(i)} = \mathbf{x}^{(i-1)}$ and $\hat{p}^J(\mathbf{y}|\boldsymbol{\theta}^{(i)}) = \hat{p}^J(\mathbf{y}|\boldsymbol{\theta}^{(i-1)})$.

2.5.3 Diagnosing MCMC convergence

MCMC methods have been successfully applied to many complex problems in statistics. However, they present a set of important drawbacks, which often hinder they application in practice [62, 147, 66, 105]:

- Firstly, it is hard to assess when the Markov chain has reached its stationary distribution, and no stopping rules have been defined that guarantee convergence to the target distribution.
- As already mentioned, they are very sensitive to the selection of the transition kernel and its variance, and prone to get stuck in local modes.
- The generation of the Markov chains is a process which cannot be

easily parallelized, since samples are processed iteratively. This also hinders the application of MCMC methods in sequential Bayesian estimation settings.

- The resulting sample presents correlations, which reduces its efficiency.

A common way of assessing the mixing and convergence properties of an MCMC method is by visual inspection of the resulting Markov chain. When it presents a noise-like appearance with a fast convergence to a stationary regime, it suggests good mixing of the chain. On the contrary, very slow variations or long constant intervals indicate that the transition kernel is not properly selected and the acceptance rate is either too high or too low, respectively. However, an assessment of stationarity based on a single chain, when the posterior pdf is multimodal can often be misleading, due to the "you've only seen where you've been" phenomenon. Generating multiple chains with different initial conditions can help to identify multimodality, but it significantly increases the computational cost [147].

To monitor the efficiency of an MCMC sampling scheme one can resort to the NESS, which is often defined differently for MCMC and IS schemes [147]. In the MCMC literature, the NESS gives the relative size of an i.i.d. sample, with the same variance as the current one, and thus indicates the loss in efficiency due to the use of a Markov chain [147]. In this case the NESS can be computed as

$$M^{neff} = \frac{1}{1 + 2\sum_{j=1}^{\infty} \hat{\rho}(j)}, \tag{2.25}$$

where $\hat{\rho}(j) = \text{corr}(\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(j)})$ is the average autocorrelation function (ACF) at lag j. For the computation of the NESS, it is common to truncate j, for example when $\hat{\rho}(j) < 0.1$.

Similarly to the degeneracy problem addressed for IS techniques, MCMC methods also suffer from instabilities and inefficiency when the target distribution concentrates in a small region of the parameter space. Note that the acceptance rate in an MCMC method depends on the ratio between the target pdf evaluated at the current and the previous samples. If either the dimension of the parameter space K, or the number of observations N is high, the acceptance probability of an MCMC method can present high variations along the iterations, yielding very low values in general. This can be avoided using a transition kernel with a very low variance, which in turn yields a poor exploration of the space of $\boldsymbol{\theta}$. In such cases, which often occur in practical problems, very large chains would be required in order to obtain

a reasonable number of accepted samples and a sufficient ESS. The same as for IS, this a clear consequence of the "curse of dimensionality".

2.6 Population Monte Carlo methods

The population Monte Carlo (PMC) algorithm [32, 30, 147] is based on an iterative version of the IS technique, and allows to approximate a static target density $\pi(\boldsymbol{\theta})$ and integrals of the form (f, π), based on sets of independent random samples in the space of $\boldsymbol{\theta}$. The PMC method aims at iteratively improving the proposal distribution used in the IS scheme so that the obtained samples better represent the target distribution and the sampling efficiency improves along the iterations. It combines the IS approach with resampling steps to yield unweighted samples if required. Thus, the PMC method produces, at each iteration, both a sample approximately distributed according to the target distribution and estimates of integrals under that distribution.

The method is named after the work of [81], which encloses the existing methods based on simultaneous generation of collections or "populations" of samples (such as SMC methods) under a family which the author calls "population Monte Carlo" algorithms, opposite to MCMC methods, that generate single samples at a time.

PMC methods are closely related to SMC and MCMC, the main families of Monte Carlo methods. As already discussed, SMC methods implement sequential IS schemes and are mainly designed for the recursive approximation of dynamical posterior distributions in state-space models, as observations are collected. On the contrary, PMC methods assume that the whole set of observations is available in advance, and implement an iterative IS scheme to approximate static posterior distributions in batch mode. Thus, both SMC and PMC have the IS technique at the core, and deal with large sets of samples or particles "simultaneously" at each time step or iteration.

PMC and MCMC techniques have the same goal, namely the approximation of static target pdfs in batch setups. However, the principle behind MCMC methods is completely different to that of PMC. MCMC algorithms generate single samples at each iteration and the convergence of such schemes to the target distribution is hard to assess.

PMC methods aim at filling the gap between the SMC and MCMC methodologies, and combine them into a coherent simulation principle. Thus PMC shares with MCMC common fields of application and can borrow from

it the construction of the proposal distribution. On the other hand, PMC uses an IS approach and resampling steps in a similar way to SMC methods.

In this section we present the generic form of the PMC algorithm and the main extensions proposed in the literature in recent years. We discuss the main features, advantages and drawbacks of this method, compared to related and alternative techniques.

2.6.1 PMC algorithm

The PMC method [32] is an iterative IS scheme that seeks to generate a sequence of proposal pdfs $q_\ell(\boldsymbol{\theta})$, $\ell = 1, \ldots, L$, such that every new proposal is closer (in some adequate sense to be defined) to the target pdf $\pi(\boldsymbol{\theta})$. In problems of the type described in Section 2.2.1, the target density is the posterior pdf of the model parameters, i.e., $\pi(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y})$.

At each iteration, $\ell = 1, \ldots, L$, the PMC method firstly selects a proposal pdf $q_\ell(\boldsymbol{\theta})$. A collection of samples $\{\boldsymbol{\theta}_\ell^{(i)}\}_{i=1}^M$ is generated from the proposal $q_\ell(\boldsymbol{\theta})$ and the associated normalized IWs $w_\ell^{(i)}$, $i = 1, \ldots, M$, are computed. A resampling step can be performed at each iteration $\ell = 1, \ldots, L$ to eliminate samples with negligible IWs and yield a set of unweighted samples $\{\tilde{\boldsymbol{\theta}}_\ell^{(i)}\}_{i=1}^M$. The generic PMC algorithm is outlined in Table 2.8.

Table 2.8: Generic PMC algorithm [32].

Iteration ($\ell = 1, \ldots, L$):

1. Select a proposal pdf $q_\ell(\boldsymbol{\theta})$:
   - If $\pi(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y})$, at iteration $\ell = 1$ the proposal may be selected as the prior $q_1(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$.
   - At iterations $\ell = 2, \ldots, L$, the proposal pdf $q_\ell(\boldsymbol{\theta})$ must be adapted according to the performance of the previous weighted sample $\{\boldsymbol{\theta}_{\ell-1}^{(i)}, w_{\ell-1}^{(i)}\}_{i=1}^M$.
2. Draw a collection of M independent samples $\{\boldsymbol{\theta}_\ell^{(i)}\}_{i=1}^M$ from $q_\ell(\boldsymbol{\theta})$.
3. Compute normalized IWs as $w_\ell^{(i)*} \propto \pi(\boldsymbol{\theta}_\ell^{(i)})/q_\ell(\boldsymbol{\theta}_\ell^{(i)})$, $w_\ell^{(i)} = w_\ell^{(i)*} / \sum_{j=1}^M w_\ell^{(j)*}$, $i = 1, \ldots, M$.

The selection of the proposal pdf $q_\ell(\boldsymbol{\theta})$ at iterations $\ell = 2, \ldots, L$ is an

essential feature of the PMC method but is not universally specified in the definition of the algorithm in [32]. As in standard IS, in order to guarantee convergence to the target pdf and finite variance estimates, the proposals should have heavier tails and a larger support than the target pdf [147]. The proposal pdf can be constructed as a mixture of Gaussian random walks, i.e.,

$$q_\ell(\boldsymbol{\theta}) = \sum_{i=1}^{M} w_{\ell-1}^{(i)} \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}_{\ell-1}^{(i)}, \boldsymbol{\Sigma}_\ell), \tag{2.26}$$

which is equivalent to resampling the set $\{\boldsymbol{\theta}_{\ell-1}^{(i)}\}_{i=1}^{M}$ according to the IWs $w_{\ell-1}^{(i)}$ and perturbing the resulting unweighted samples $\{\tilde{\boldsymbol{\theta}}_{\ell-1}^{(i)}\}_{i=1}^{M}$ with Gaussian noise. The covariance matrix of the transition kernel $\boldsymbol{\Sigma}_\ell$ can be adapted according to previous performances. The transition kernel can itself be a mixture of components with different scales, as in [32, 52]. Random walk proposals are more open to complex settings, and are often preferred for high-dimensional targets [52, 80]. However, similarly to MCMC techniques, they can lead to poor exploration of the parameter space, specially when the target presents multimodality.

Alternatively, the proposal can also be constructed based on the moments of the previous weighted sample $\{\boldsymbol{\theta}_{\ell-1}^{(i)}, w_{\ell-1}^{(i)}\}_{i=1}^{M}$ [30]. The proposals could also include a heavy tailed component as in defensive sampling [30], or could be built as a nonparametric kernel approximation to $\pi(\boldsymbol{\theta})$ [32]. The proposal may depend both on the sample index i and the iteration ℓ, and possibly on all the previously generated samples, if sufficient storage is available. However, in this work we restrict the definition of the algorithm to the scheme in Table 2.8.

When the target pdf can be evaluated exactly, the PMC algorithm retains the unbiasedness property of the IS technique and the IWs are naturally normalized. On the contrary, if it can only be evaluated up to a normalizing constant, the unbiasedness property only holds asymptotically in M and the weights have to be normalized to sum up to one [32, 147].

At every iteration of the algorithm it is possible to construct a discrete approximation of the posterior distribution as

$$\pi_\ell^M(d\boldsymbol{\theta}) = p_\ell^M(d\boldsymbol{\theta}|\mathbf{y}) = \sum_{i=1}^{M} w_\ell^{(i)} \delta_{\boldsymbol{\theta}_\ell^{(i)}}(d\boldsymbol{\theta})$$

and compute an estimate of the integral of interest (f, π) as

$$(f, \pi_\ell^M) = \sum_{i=1}^{M} w_\ell^{(i)} f(\boldsymbol{\theta}_\ell^{(i)}).$$

If the proposals $q_\ell(\boldsymbol{\theta})$ are actually improved across iterations, it can be expected that the approximation error $|(f, \pi_\ell^M) - (f, \pi)|$ also decreases with ℓ. The convergence of the original PMC scheme is easily justified by the convergence of the standard IS method. Indeed, it can be proved [65] that the discrete measure $\pi_\ell^M(d\boldsymbol{\theta})$ converges to $\pi(\boldsymbol{\theta})d\boldsymbol{\theta}$ under mild assumptions, meaning that

$$\lim_{M\to\infty} |(f, \pi_\ell^M) - (f, \pi)| = 0 \quad \text{almost surely (a.s.)}$$

for every $\ell \in \{1, ..., L\}$ and any $f \in B(\mathbb{R}^K)$, where $B(\mathbb{R}^K)$ is the set of bounded[1] real functions over $\mathbb{R}^K$.

The PMC algorithm provides a set of important advantages w.r.t. its MCMC counterpart. Given that sets of M independent samples are processed at each iteration, most of the computations of the PMC method can be easily paralellized, drastically reducing its computation time. Only the normalization of the IWs and the proposal update step must be performed in a centralized manner, since they require all samples and weights. Additionally, and contrary to MCMC methods, PMC yields independent samples and asymptotically unbiased estimates at each iteration, which avoids the need of a convergence period. However, the algorithm is well known to suffer from the curse of dimensionality [19, 107]. In [107] the performance of a single step of the PMC algorithm is analyzed in high-dimensional problems. The authors demonstrate that the asymptotic variance of the estimates grows exponentially with the dimensionality of the parameter space.

In Figure 2.5 we show an example of the performance of the PMC algorithm, which generates a sequence of proposal distributions that approach the target pdf along the iterations.

Degeneracy of IWs

The main limitation of the PMC method is the already mentioned problem of degeneracy of the IWs, common to all IS-based techniques. Ideally, we would expect the ESS to increase along the iterations of the PMC algorithm

[1] In particular, $f \in B(\mathbb{R}^K)$ if, and only if, $\|f\|_\infty = \sup_{\boldsymbol{\theta}\in\mathbb{R}^K} |f(\boldsymbol{\theta})| < \infty$.

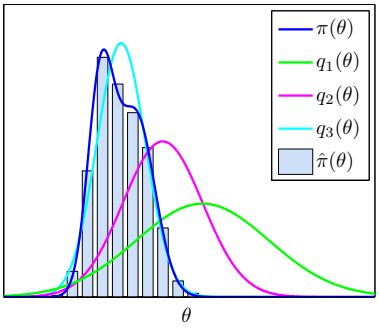

Figure 2.5: Illustration of the performance of the PMC algorithm with Gaussian proposals.

as it converges, yielding better estimates. However, except for very simple and low-dimensional problems, the normalized IWs computed in step 3 of the PMC method present extreme variations, yielding an ESS close to 1. This hinders the proposal update at the next iteration, and often causes numerical problems during the execution of the algorithm. For this reason, IS and PMC methods are usually avoided in practical applications where MCMC methods can be applied [47].

The degeneracy problem is particularly severe at the first iterations of the algorithm, since the prior knowledge is usually vague and the initial proposals can be very broad in comparison with the target pdf. The resampling technique has been proposed to mitigate the degeneracy problem arising in sequential IS setups, where sample impoverishment is only slight or moderate, depending on the dimension of the problem. In fact, resampling is often required only in some sequential steps, when the ESS drops below a threshold, say $J/2$. In sequential setups the target posterior distributions (for instance, in tracking applications) changes relatively slowly and samples approximating the posterior at time n allow to predict reasonably well the posterior at time $n+1$. Observations are gathered in small sets, which results in less variations of the IWs, and a higher ESS. However, in iterative IS schemes, all the observations are available in a batch and the degeneracy of the IWs is so severe that the ESS usually barely exceeds a value of 1, even in very simple problems. For this reason, the resampling step does not avoid degeneracy in this kind of problems and several methods that try to flatten the target pdf have been proposed [41, 47, 20].

2.6.2 D-kernel PMC

A popular, and often effective, approach to the construction of proposal pdfs is to build them as mixtures of random walks [32, 52, 53, 80]. The D-kernel PMC (DPMC) method proposed in [32, 52] constructs the proposal densities as mixtures of D Gaussian random walks centered at the previous unweighted samples $\tilde{\boldsymbol{\theta}}_{\ell-1}^{(i)}$, namely,

$$q_\ell(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}_{\ell-1}^{(i)}) = \sum_{d=1}^{D} \alpha_{\ell,d}\, \mathcal{N}(\boldsymbol{\theta}; \tilde{\boldsymbol{\theta}}_{\ell-1}^{(i)}, \sigma_d^2 \mathbf{I}), \quad \sum_{d=1}^{D} \alpha_{\ell,d} = 1,$$

where the scales σ_d^2 are fixed and must be set a priori. The mixture weights $\alpha_{\ell,d}$ are adapted along the iterations as the relative importance of each kernel in the mixture, i.e.,

$$\alpha_{\ell+1,d} = \sum_{i=1}^{M} w_\ell^{(i)} I_d(Z_\ell^{(i)}),$$

where $Z_\ell^{(i)} \in \{1, \ldots, D\}$ is the random index that identifies from which kernel the sample $\boldsymbol{\theta}_\ell^{(i)}$ has been drawn and $I_d(Z_\ell^{(i)})$ is an indicator function, which is equal to 1 if $Z_\ell^{(i)} = d$ and 0 otherwise.

The DPMC algorithm proposed in [32, 52] is shown in Table 2.9. The authors of [52] prove that this algorithm allows to minimize, along the iterations, the Kullback-Leibler divergence (KLD) between the proposal pdf q_ℓ and the target pdf π, i.e.,

$$\mathcal{D}(\pi \| q_\ell) = \int \pi(\boldsymbol{\theta}) \log \frac{\pi(\boldsymbol{\theta})}{q_\ell(\boldsymbol{\theta})} d\boldsymbol{\theta}. \tag{2.27}$$

In [53], a version of this algorithm was developed in which the adaptation of the mixture weighs minimizes the asymptotic variance of the IS procedure.

The proposal construction in the DPMC algorithm is actually equivalent to sampling from a double mixture of the form

$$q_\ell(\boldsymbol{\theta}) = \sum_{i=1}^{M} w_{\ell-1}^{(i)} \sum_{d=1}^{D} \alpha_{\ell,d}\, \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}_{\ell-1}^{(i)}, \sigma_d^2 \mathbf{I}),$$

based on the previous set of samples and IWs $\{\boldsymbol{\theta}_{\ell-1}^{(i)}, w_{\ell-1}^{(i)}\}$, thus taking into account the resampling step in the computation of the IWs. In [80] a PMC scheme based on the DPMC method of [30, 52] is proposed, which computes the IWs based on this construction of the proposal pdf, thus eliminating the dependence of $w_\ell^{(i)}$ on the previous sample point $\boldsymbol{\theta}_{\ell-1}^{(i)}$ and reducing their variations.

Table 2.9: D-kernel PMC algorithm [32, 52].

Iteration $\ell = 1$:

1. Draw a collection of M samples $\{\boldsymbol{\theta}_1^{(i)}\}_{i=1}^M$ from $q_1(\boldsymbol{\theta})$.
2. Compute the normalized IWs as $w_1^{(i)} \propto \pi(\boldsymbol{\theta}_1^{(i)})/q_1(\boldsymbol{\theta}_1^{(i)})$.
3. Resample with replacement the set $\{\boldsymbol{\theta}_1^{(i)}\}_{i=1}^M$ according to the weights $w_1^{(i)}$ to obtain $\{\tilde{\boldsymbol{\theta}}_1^{(i)}\}_{i=1}^M$.
4. Set the mixture weights for iteration $\ell = 2$ to $\alpha_{2,d} = 1/D$, $d = 1, \ldots, D$.

Iterations $\ell = 2, \ldots, L$:

1. Draw M samples $\{\boldsymbol{\theta}_\ell^{(i)}\}_{i=1}^M$ from the D-kernel proposal centered at the previous unweighted samples and with mixture weights $\alpha_{\ell,d}$, $d = 1, \ldots, D$,

$$q_\ell(\boldsymbol{\theta}|\tilde{\boldsymbol{\theta}}_{\ell-1}^{(i)}) = \sum_{d=1}^{D} \alpha_{\ell,d} \mathcal{N}(\boldsymbol{\theta}; \tilde{\boldsymbol{\theta}}_{\ell-1}^{(i)}, \sigma_d^2 \mathbf{I}), \quad i = 1, \ldots, M.$$

2. Compute the normalized IWs as $w_\ell^{(i)} \propto \pi(\boldsymbol{\theta}_\ell^{(i)})/q_\ell(\boldsymbol{\theta}_\ell^{(i)}|\tilde{\boldsymbol{\theta}}_{\ell-1}^{(i)})$.
3. Resample with replacement the set $\{\boldsymbol{\theta}_\ell^{(i)}\}_{i=1}^M$ according to the weights $w_\ell^{(i)}$ to obtain $\{\tilde{\boldsymbol{\theta}}_\ell^{(i)}\}_{i=1}^M$.
4. Update the mixture weights $\alpha_{\ell,d}$, $d = 1, \ldots, D$, as

$$\alpha_{\ell+1,d} = \sum_{i=1}^{M} w_\ell^{(i)} I_d(Z_\ell^{(i)}).$$

2.6.3 Mixture PMC

A powerful extension of the DPMC, termed mixture PMC (MPMC), was proposed in [30], which constructs the sequence of proposal pdfs as mixtures of multivariate Gaussian pdfs of the form

$$q_\ell(\boldsymbol{\theta}) = \sum_{d=1}^{D} \alpha_{\ell,d} q_{\ell,d}(\boldsymbol{\theta}; \boldsymbol{\beta}_{\ell,d}), \tag{2.28}$$

where both the mixture weights $\alpha_{\ell,d}$ and the kernel parameters $\boldsymbol{\beta}_{\ell,d}$ of each component are adapted along the iterations minimizing the KLD between the target and the proposal pdf. This construction of the proposal pdf allows to approximate essentially any multimodal target distribution, as long as an upper bound D for the number of components is available. This is in contrast with the main existing MCMC techniques, which are not appropriate for approximating multimodal target distributions because they are prone to get stuck in local modes [147, 66]. The MPMC update mechanism is similar to the expectation-maximization (EM) algorithm [23] with the E-step replaced by IS computations, and is outlined in Table 2.10.

The updating of the internal kernel parameters can often lead to challenging robustness problems, particularly in high dimension. The authors of [30] propose a Rao-Blackwellization (RB) scheme that empirically shows to be very effective to fight against these numerical issues, with a low additional computational complexity. It consists of replacing the index $Z_\ell^{(i)}$, that identifies from which kernel the i-th sample has been drawn, by its conditional expectation given $\boldsymbol{\theta}_\ell^{(i)}$, i.e., $\rho_{\ell,d}^{(i)}$, defined in equation (2.29). Thus, in the RB scheme, each mixture component is updated based on all samples $\boldsymbol{\theta}_\ell^{(i)}$, rather than only on those generated from the given component. The plain and RB schemes only differ in step 3.

The MPMC method has been particularized for the Gaussian and Student's t mixture cases, providing expressions for the update of the parameters, that are given below [30].

Gaussian mixture importance functions

Assume that the proposal pdf $q_\ell(\boldsymbol{\theta})$ at iteration ℓ is a mixture of D, K-dimensional Gaussian kernels of the form

$$q_{\ell,d}(\boldsymbol{\theta}; \boldsymbol{\beta}_{\ell,d}) = \mathcal{N}_K(\boldsymbol{\theta}; \boldsymbol{\mu}_{\ell,d}, \boldsymbol{\Sigma}_{\ell,d}), \quad d = 1, \ldots, D,$$

where $\boldsymbol{\mu}_{\ell,d}$ and $\boldsymbol{\Sigma}_{\ell,d}$ are the mean vector and covariance matrix of each component, respectively. These parameters are updated for the next

Table 2.10: Mixture PMC algorithm [30].

Iteration ($\ell = 1, \ldots, L$):

1. Generate a sample $\{\boldsymbol{\theta}_\ell^{(i)}\}_{i=1}^M$ from the current mixture proposal $q_\ell(\boldsymbol{\theta})$ in equation (2.28).

2. For $i = 1, \ldots, M$, compute normalized IWs $w_\ell^{(i)} \propto \pi(\boldsymbol{\theta}_\ell^{(i)})/q_\ell(\boldsymbol{\theta}_\ell^{(i)})$ and normalized mixture posterior probabilities $\rho_{\ell,d}^{(i)}$, which satisfy $\sum_{d=1}^D \rho_{\ell,d}^{(i)} = 1$, as

$$\rho_{\ell,d}^{(i)} \propto \alpha_{\ell,d} q_{\ell,d}(\boldsymbol{\theta}_\ell^{(i)}; \boldsymbol{\beta}_{\ell,d}). \tag{2.29}$$

3. Update the weights and the parameters of each component as

$$\alpha_{\ell+1,d} = \sum_{i=1}^M w_\ell^{(i)} \xi_{\ell,d}^{(i)} \quad \text{and} \tag{2.30}$$

$$\boldsymbol{\beta}_{\ell+1,d} = \arg\max_{\boldsymbol{\beta}_{\ell,d}} \left[\sum_{i=1}^M w_\ell^{(i)} \xi_{\ell,d}^{(i)} \log q_{\ell,d}(\boldsymbol{\theta}_\ell^{(i)}; \boldsymbol{\beta}_{\ell,d}) \right]. \tag{2.31}$$

   In the plain MPMC scheme, $\xi_{\ell,d}^{(i)} = I_d(Z_\ell^{(i)})$, while in the RB-MPMC scheme $\xi_{\ell,d}^{(i)} = \rho_{\ell,d}^{(i)}$.

iteration $\ell + 1$ as [30]

$$\boldsymbol{\mu}_{\ell+1,d} = \frac{\sum_{i=1}^M w_\ell^{(i)} \rho_{\ell,d}^{(i)} \boldsymbol{\theta}_\ell^{(i)}}{\alpha_{\ell+1,d}} \quad \text{and}$$

$$\boldsymbol{\Sigma}_{\ell+1,d} = \frac{\sum_{i=1}^M w_\ell^{(i)} \rho_{\ell,d}^{(i)} (\boldsymbol{\theta}_\ell^{(i)} - \boldsymbol{\mu}_{\ell+1,d})(\boldsymbol{\theta}_\ell^{(i)} - \boldsymbol{\mu}_{\ell+1,d})^\top}{\alpha_{\ell+1,d}}.$$

Student's t mixture importance functions

The t mixture has been suggested for importance sampling, opposite to the Gaussian mixture, because its heavier tails may capture a wider range of non-Gaussian targets with a smaller number of components [30].

Thus, assume that the proposal pdf at iteration ℓ is a mixture of D, K-dimensional Student's t kernels (with a fixed number ν_d of degrees of freedom) denoted

$$q_{\ell,d}(\boldsymbol{\theta};\boldsymbol{\beta}_{\ell,d}) = \mathcal{T}_K(\boldsymbol{\theta};\boldsymbol{\mu}_{\ell,d},\boldsymbol{\Sigma}_{\ell,d},\nu_d), \quad d=1,...,D.$$

We compute $\gamma_{\ell,d}^{(i)} = \frac{\nu_d+K}{\nu_d+(\boldsymbol{\theta}_\ell^{(i)}-\boldsymbol{\mu}_{\ell,d})^\top \boldsymbol{\Sigma}_{\ell,d}^{-1}(\boldsymbol{\theta}_\ell^{(i)}-\boldsymbol{\mu}_{\ell,d})}$ and the mean and covariance parameters are updated as [30]

$$\boldsymbol{\mu}_{\ell+1,d} = \frac{\sum_{i=1}^M w_\ell^{(i)} \rho_{\ell,d}^{(i)} \gamma_{\ell,d}^{(i)} \boldsymbol{\theta}_\ell^{(i)}}{\sum_{i=1}^M w_\ell^{(i)} \rho_{\ell,d}^{(i)} \gamma_{\ell,d}^{(i)}} \quad \text{and}$$

$$\boldsymbol{\Sigma}_{\ell+1,d} = \frac{\sum_{i=1}^M w_\ell^{(i)} \rho_{\ell,d}^{(i)} \gamma_{\ell,d}^{(i)} (\boldsymbol{\theta}_\ell^{(i)}-\boldsymbol{\mu}_{\ell+1,d})(\boldsymbol{\theta}_\ell^{(i)}-\boldsymbol{\mu}_{\ell+1,d})^\top}{\alpha_{\ell+1,d}}.$$

2.6.4 Other extensions and related techniques

In our opinion, the MPMC algorithm of [30] has been the most significant advance in the design of generic (non problem specific) proposal functions for PMC algorithms. However, some other extensions have been proposed in the literature. Here, we review the main contributions to this topic and some related techniques.

In [36] a PMC scheme for state-space, or missing data, models was introduced. The proposed method mimics the Gibbs sampler by drawing the $\boldsymbol{\theta}$'s and $\mathbf{x}$'s from their conditional distributions with pdfs $p(\boldsymbol{\theta}|\mathbf{x},\mathbf{y})$ and $p(\mathbf{x}|\boldsymbol{\theta},\mathbf{y})$, respectively, which requires that both densities are known up to a normalizing constant. This method is also exposed to degeneracy and, to partly alleviate this problem, the authors of [36] propose to apply a RB technique.

In [28] a marginalized PMC method is introduced that can improve the efficiency of the sampling scheme in problems where some of the parameters are conditionally linear and can be analytically integrated. This approach is related to the RB technique. An extension of the marginalized PMC method for high-dimensional problems is proposed in [149], termed multiple marginalized PMC (MultiPMC). This algorithm splits the multidimensional parameter space into several subspaces of lower dimension, which are handled by a set of marginalized PMC estimators in a distributed manner. The different PMC blocks may exchange information regarding importance functions, samples and weights. The

partitioning depends on the particular problem and, when feasible, it allows to significantly reduce the computational complexity of the algorithm.

In [78] a PMC algorithm for the joint model selection and parameter estimation is introduced. It assumes that a given model out of a finite set $r = 1, \ldots, R$, and with associated parameters $\boldsymbol{\theta}_r$, has originated the observations. The goal is to identify the true model and to estimate its parameters, i.e., the target pdf is given by $p(r, \boldsymbol{\theta}|\mathbf{y})$. The proposed method includes a two-stage sampling scheme, for the model index r and the parameters $\boldsymbol{\theta}_r$, respectively, which allows to sample from parameter spaces with different dimensions.

In [50] a PMC algorithm is proposed that focuses on an efficient sampling procedure from the proposal distribution q_ℓ in high-dimensional problems. This method draws samples from proposal distributions conditional on all-but-one components of the previous sample. It is similar to the Gibbs sampling method, except that the conditionals are not constructed from the target distribution, which is usually intractable. Thus, at each iteration $\ell = 1, \ldots, L$ and for $i = 1, \ldots, M$, new samples are generated as $\theta_{\ell,k}^{(i)} \sim q(\theta_k|\boldsymbol{\theta}_{\ell-1,\backslash k}^{(i)})$, $k = 1, \ldots, K$, where $\boldsymbol{\theta}_{\backslash k}$ denotes the vector containing all parameters in $\boldsymbol{\theta}$ except for θ_k. The overall proposal distribution is a product of conditionals, which allows for an efficient sampling and evaluation. However, the resulting IWs can still present severe degeneracy due to the extreme values of the likelihood function in high-dimensional spaces. In [150] the marginalized PMC method is combined with Gibbs-based PMC sampling to estimate multimodal posterior distributions.

The adaptive multiple IS (AMIS) method proposed in [43] is another iterative IS scheme, in which the IWs of all past and present samples are recomputed at each iteration. It is a well known fact that the major drawback of adaptive IS algorithms is that the initial proposal distribution has a big influence on the performance of the method and it is usually hard to recover from a poor initial sample. For this reason, the authors of [43] suggest to invest a major part of the computation effort on the initialization stage. This method has a significantly higher computational complexity than the conventional PMC algorithm, and its convergence properties have not been investigated analytically.

The PMC algorithm can also be interpreted as a particular case of the SMC sampler of [47], in which the target distribution does not change along iterations, i.e., $\pi_\ell(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta})$, for $\ell = 1, \ldots, L$. The PMC method is also related to the IBIS method of [41], in what it considers iterated IS with changing proposals.

2.7 Approximate Bayesian computation

All model-based statistical methods for the approximation of posterior distributions $p(\boldsymbol{\theta}|\mathbf{y})$ in the Bayesian framework rely on the possibility of evaluating the likelihood function $p(\mathbf{y}|\boldsymbol{\theta})$ [155]. For simple models, the likelihood can often be derived analytically. However, in many complex problems of great interest arising, specially, in the biological sciences, the likelihood function proves intractable or is computationally very costly to evaluate [16, 14]. These settings often require complex stochastic modeling, with high-dimensional data and parameter spaces that prevent the implementation of likelihood-based statistical inference methods.

Approximate Bayesian computation (ABC), originally proposed in [143], is the adopted label for a class of computational algorithms that allow to perform Bayesian inference avoiding the evaluation of the likelihood function. Assuming that generation of samples under the observation model $p(\mathbf{y}|\boldsymbol{\theta})$ is feasible, the basic form of the ABC method proceeds as follows: a candidate parameter value $\boldsymbol{\theta}^\star$ is drawn from the prior distribution $p(\boldsymbol{\theta})$. Then, an auxiliary sample $\mathbf{y}^\star$ with the same dimension as $\mathbf{y}$ is drawn from the observation model, i.e., $\mathbf{y}^\star \sim p(\mathbf{y}|\boldsymbol{\theta}^\star)$. The simulated and the observed data are compared in terms of some distance measure denoted $\rho(\mathbf{y}, \mathbf{y}^\star)$, and the candidate $\boldsymbol{\theta}^\star$ is accepted as a sample from the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$ if the computed distance is small enough, say $\rho(\mathbf{y}, \mathbf{y}^\star) \leq \epsilon$, for some small threshold $\epsilon > 0$. This basic scheme is known as the ABC rejection algorithm and is outlined in Table 2.11.

Table 2.11: ABC rejection algorithm [143, 16].

For $i = 1, \ldots, M$: repeat these steps until each sample $\boldsymbol{\theta}^{(i)}$ is accepted:

1. Draw a candidate $\boldsymbol{\theta}^\star \sim p(\boldsymbol{\theta})$ from the prior distribution.
2. Draw a sample $\mathbf{y}^\star \sim p(\mathbf{y}|\boldsymbol{\theta}^\star)$ from the observation model.
3. Accept $\boldsymbol{\theta}^\star$ as a sample from $p(\boldsymbol{\theta}|\mathbf{y})$ if $\rho(\mathbf{y}, \mathbf{y}^\star) < \epsilon$, setting $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^\star$.

This method yields a set of samples $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^M$ approximately distributed from the posterior distribution of interest, which can be approximated as

$$\hat{p}^M(d\boldsymbol{\theta}|\mathbf{y}) = \frac{1}{M}\sum_{i=1}^{M} \delta_{\boldsymbol{\theta}^{(i)}}(d\boldsymbol{\theta}).$$

For reasons of computational tractability the distance function ρ is often defined in terms of summary statistics, $S(\mathbf{y})$ and $S(\mathbf{y}^\star)$, of the simulated and observed data, respectively [16, 155]. Ideally, summary statistics should be sufficient for the parameter $\boldsymbol{\theta}$, i.e., they should provide as much information about the parameter $\boldsymbol{\theta}$ as the data itself. If the distance function ρ and the threshold value ϵ are selected properly, the posterior pdf of interest at sample $\boldsymbol{\theta}^\star$ can be approximated as

$$p(\boldsymbol{\theta}^\star|\mathbf{y}) = p(\boldsymbol{\theta}^\star|S(\mathbf{y})) \approx p(\boldsymbol{\theta}^\star|\rho(S(\mathbf{y}), S(\mathbf{y}^\star)) \leq \epsilon).$$

This approximation improves as ϵ decreases, yielding exact values as $\epsilon \to 0$ if $S(\cdot)$ is sufficient for $\boldsymbol{\theta}$. However, sufficient summary statistics are generally unavailable and the use of non-sufficient statistics is common in practice [14], which leads to a loss in the accuracy of the approximation. The tolerance parameter ϵ determines the trade-off between the acceptance rate and the accuracy of the approximation, and is hard to set in practice, yet for small values of ϵ, the rejection rate can be extremely high [155].

In Figure 2.6 the principle of the ABC rejection algorithm is illustrated. In the *left* plot it can be observed that the target pdf is accurately approximated as long as the number of accepted samples is sufficiently large. However, the rejected samples constitute a major part of the sample set (note that both histograms are normalized). The plot on the *right* depicts the sampling procedure: pairs of samples $\theta^\star, y^\star$ are drawn from the prior distribution and the observation model, respectively. Those pairs with a distance $\rho(\mathbf{y}, \mathbf{y}^\star)$ to the real observations y below a certain threshold ϵ, are accepted. It is clear that the lower the threshold ϵ, the lower the amount of accepted samples.

Multiple extensions of the ABC principle have been proposed in the literature [155], which can be classified into two big groups. On the one hand, MCMC-ABC algorithms [122] explore the parameter space iteratively using Markov chains with the desired stationary distribution. On the other hand, SMC-ABC methods [151, 48] perform IS, generating a large set of candidates at each iteration according to a transition kernel and computing IWs associated to those samples. In this section we briefly introduce the idea behind MCMC-ABC algorithms as well as a particular case of the SMC-ABC method, termed PMC-ABC algorithm and proposed in [15]. The latter technique has been shown to yield the best performance among different SMC-ABC methods [155].

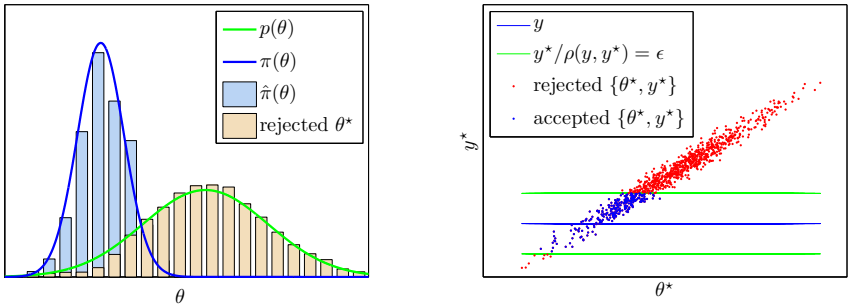

Figure 2.6: Illustrative example of the ABC rejection algorithm. The proposal and the target distribution are shown in the *left* plot, together with the normalized histogram of the accepted and the rejected samples. The *right* plot illustrates the sampling procedure and the acceptance criterion.

2.7.1 MCMC-ABC algorithm

ABC and MCMC computations can be easily combined [122, 155]. Focusing on the MH algorithm described in Table 2.6, the MH-ABC algorithm proceeds as follows: at iteration i, a candidate $\boldsymbol{\theta}^\star$ is drawn from a proposal density $q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(i-1)})$, and the associated observation is simulated as $\mathbf{y}^\star \sim p(\mathbf{y}|\boldsymbol{\theta}^\star)$. Then, the acceptance probability α for sample $\boldsymbol{\theta}^\star$ is computed as

$$\alpha = \begin{cases} \min\left\{1, \frac{p(\boldsymbol{\theta}^\star)}{p(\boldsymbol{\theta}^{(i-1)})} \times \frac{q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^\star)}{q(\boldsymbol{\theta}^\star|\boldsymbol{\theta}^{(i-1)})}\right\} & \text{if } \rho(\mathbf{y}, \mathbf{y}^\star) \leq \epsilon \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the candidate sample $\boldsymbol{\theta}^\star$ is always rejected if the corresponding observation $\mathbf{y}^\star$ is not similar enough to the true observation $\mathbf{y}$. On the other hand, it can be accepted with a given probability if the ABC condition is satisfied, setting $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^\star$. Otherwise, the previous sample is replicated, i.e., $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$.

The ABC principle can be equally embedded into other MCMC algorithms [155]. However, the resulting MCMC-ABC schemes are particularly likely to get stuck or yield extremely high rejection rates, requiring a prohibitive computational cost even for simple problems. The partial rejection control (PRC)-ABC method was developed in [151] as an alternative to MCMC-ABC methods, which suffer from severe mixing problems.

2.7.2 PMC-ABC algorithm

The PMC-ABC method was proposed in [15] as an alternative to the PRC-ABC method, which has been shown to introduce a bias in the approximation of the posterior [15, 155]. The PMC-ABC algorithm is inspired in the iterative IS procedure of the PMC algorithm, and avoids the evaluation of the likelihood function by the application of the ABC principle. This method requires the selection of a sequence of proposal pdfs $q_\ell(\boldsymbol{\theta})$, $\ell = 1, \ldots, L$, and a sequence of decreasing tolerance thresholds $\epsilon_1 \geq \ldots \geq \epsilon_L$. The PMC-ABC algorithm is shown in Table 2.12.

Table 2.12: PMC-ABC algorithm [15].

Iterations $\ell = 1, \ldots, L$:

1. Select a proposal distribution $q_\ell(\boldsymbol{\theta})$:
   - at iteration $\ell = 1$, let $q_1(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$,
   - at iterations $\ell = 2, \ldots, L$, select $q_\ell(\boldsymbol{\theta})$ according to the previous set of weighted samples $\{\boldsymbol{\theta}_{\ell-1}^{(i)}, w_{\ell-1}^{(i)}\}_{i=1}^{M}$.
2. For $i = 1, \ldots, M$, simulate $\boldsymbol{\theta}_\ell^{(i)} \sim q_\ell(\boldsymbol{\theta})$ and $\mathbf{y}_\ell^{(i)} \sim p(\mathbf{y}|\boldsymbol{\theta}_\ell^{(i)})$ until $\rho(\mathbf{y}, \mathbf{y}_\ell^{(i)}) \leq \epsilon_\ell$.
3. Compute normalized IWs as $w_\ell^{(i)} \propto p(\boldsymbol{\theta}_\ell^{(i)})/q_\ell(\boldsymbol{\theta}_\ell^{(i)})$.

The authors of [15] propose to construct the proposal pdf at iterations $\ell = 2, \ldots, L$ as the mixture of Gaussian random walks in equation (2.26), where the convariance matrix is constructed as $\boldsymbol{\Sigma}_\ell = \sigma_\ell^2 \mathbf{I}$ and σ_ℓ^2 is set to twice the empirical variance of the weighted set at the previous iteration [15]. This algorithm optimizes the acceptance probability and minimizes the KLD between the target and the proposal pdf [155]. In addition, it requires the setting of the fewest tuning parameters among ABC methods. However, the efficiency of this sampling scheme is still very low.

2.8 Summary

In this chapter we have briefly reviewed the basics of the Bayesian inference methodology, which combines a prior knowledge on the variables of interest

with the available observations to yield the desired posterior distribution. We have addressed the problem of Bayesian inference both in static and dynamical models, by means of Monte Carlo approximations. Monte Carlo methods allow to simulate from complex distributions and approximate integrals w.r.t. them. We have presented the basic Monte Carlo method, which is seldom applicable in practice, and the fairly universal importance sampling (IS) technique, which is the basis of the methods investigated in this thesis.

We have discussed the main families of Monte Carlo methods which are relevant for the present work, either as tools employed by the proposed algorithms or for comparison purposes. Among them, sequential Monte Carlo (SMC) methods allow to approximate posterior distributions is state-space models. Markov chain Monte Carlo (MCMC) methods are the main tool for the posterior approximation of static parameters. The population Monte Carlo methodology is an alternative to MCMC methods with interesting features, but it is rarely used when an MCMC method can be applied, given its inefficiency in high-dimensional problems. Finally, we have presented the core ideas behind ABC, or likelihood-free, methods.

Chapter 3

Nonlinear population Monte Carlo algorithms

In this chapter we introduce a new family of PMC algorithms for Bayesian inference. In Section 3.1 we motivate the problem and describe the problematic of Bayesian inference in high-dimensional spaces. In Section 3.2 we describe the nonlinear IS (NIS) technique, which performs nonlinear transformation to the IWs in order to alleviate the degeneracy problem. In Section 3.3 we introduce a nonlinear PMC (NPMC) method, which is based on iterative NIS and constructs the proposal pdf as a Gaussian distribution. As a generalization for arbitrary multimodal distributions, in Section 3.4 we propose a nonlinear MPMC (NMPMC) method to extend the MPMC algorithm of [30]. In Section 3.5 we address the Monte Carlo approximation of posterior distributions in state-space models. Section 3.6 summarizes the connections of the proposed NPMC algorithms to existing techniques. Finally, Section 3.7 is devoted to the conclusions of this chapter.

3.1 Importance sampling in high dimension

In Chapter 2 we have introduced and discussed the main existing Monte Carlo methods for the approximation of posterior distributions in the Bayesian framework. In this work we have developed a family of offline Monte Carlo methods, which process the available observations in batch mode. We have addressed the static inference problem described in Section 2.2.1, where the target distribution of interest is a posterior distribution with density $\pi(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y})$, of a set of random parameters $\boldsymbol{\theta}$ given some observed data $\mathbf{y}$. Additionally, we have considered the approximation of the

joint posterior pdf $p(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$ of the parameters and hidden states in state-space models, which has been introduced in Section 2.2.2. The methods to be described can be applied in a similar manner to the general case where the target distribution is not necessarily a posterior distribution.

As already discussed in Chapter 2, the generation of samples that represent the pdfs $p(\boldsymbol{\theta}|\mathbf{y})$ and $p(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$ adequately when the dimension of the parameters, the hidden states or the observations is large is normally a difficult task. Maybe contrary to intuition, when a large number of observations is available, and specially when they present low variance, an extremely peaky likelihood function prevents from standard techniques to perform properly. IS based methods have been traditionally avoided due to their inefficiency in these scenarios. The number of samples required by an IS technique turns out to be exceedingly large as the dimension increases and the selection of an appropriate importance function appears impossible in practical problems [147, 19]. For this reason, MCMC methods have become the standard solution to this problem [66, 6]. However, these techniques also perform inefficiently in high-dimensional spaces, yielding high rejection rates and requiring extremely long chains to attain reasonable results. Additionally, the MCMC methodology has a number of important drawbacks already discussed in Chapter 2, which have inspired the further development of alternative algorithms.

In this work, we focus on the population Monte Carlo (PMC) methodology [32] as a powerful alternative to MCMC methods with interesting features, such as sample independence, unbiasedness, and ease of parallelization. The effort in the field of PMC algorithms has been typically directed toward the design of efficient proposal functions [30, 50]. Alternatively, in this chapter we present a family of novel PMC methods, termed nonlinear PMC (NPMC). The emphasis is not placed on the proposal update scheme, which can be very simple. The main feature of the technique is the application of a nonlinear transformation to the IWs in order to reduce their variations. We thus call the modified IS approach nonlinear IS (NIS). In this way, the efficiency of the sampling scheme is improved (specially when drawing from poor proposals) and the degeneracy of the IWs is drastically mitigated even when the number of generated samples is relatively small. As a consequence, the proposed NPMC algorithms allow to approximate high-dimensional distributions and integrals with a comparatively low computational complexity. The proposed NIS technique can also be successfully combined with existing PMC algorithms to drastically improve their efficiency, yielding powerful and generic inference tools, which outperform existing state of the art techniques.

3.2 Nonlinear importance sampling

In this section we describe the nonlinear IS (NIS) method, which is the basis of the family of algorithms proposed in this thesis. The idea behind NIS is to smooth the variations of the IWs, with the aim of increasing the resulting ESS and coping with the inefficiency arising from a poor selection of the proposal distribution.

In particular, we propose a modification of the standard IS approach, which additionally computes a set of transformed IWs (TIWs) $\bar{w}^{(i)}$ associated to each sample $\boldsymbol{\theta}^{(i)}$, as a nonlinear transformation of the standard unnormalized IW $w^{(i)*}$. To be specific, one chooses a transformation function $\varphi^M : (\mathbb{R}^+)^M \times \{1,\dots,M\} \to \mathbb{R}^+$ and then computes the unnormalized TIWs as

$$\bar{w}^{(i)*} = \varphi^M(w^{(i)*}),\ i = 1,\dots,M,$$

where $w^{(i)*}$ is the standard unnormalized IW associated to $\boldsymbol{\theta}^{(i)}$ and $\varphi^M(w^{(i)*})$ is shorthand for $\varphi^M(\{w^{(j)*}\}_{j=1}^M, i)$. That is, φ^M can be a function of both the complete weight set $\{w^{(j)*}\}_{j=1}^M$ and the index i of the weight to be transformed. The TIWs are subsequently normalized to yield $\sum_{i=1}^M \bar{w}^{(i)} = 1$.

The nonlinearity φ^M should be chosen so as to reduce the variation of the normalized TIWs. Intuitively, it should preserve the ordering of the samples (those with larger IWs should also have the largest TIWs) while reducing the difference $\max_i \bar{w}^{(i)} - \min_i \bar{w}^{(i)}$ or some other measure of weight variation. This modification of the algorithm mitigates the sensitivity of the conventional IS to the selection of the proposal pdf. If the nonlinearity φ^M is selected properly, the ESS of the weighted sample $\{\boldsymbol{\theta}^{(i)}, \bar{w}^{(i)}\}_{i=1}^M$ should be significantly higher than that of the original sample with standard IWs. The NESS computed from the TIWs $\bar{w}^{(i)}$ is denoted as $\bar{M}^{neff} = [M\sum_{i=1}^M (\bar{w}^{(i)})^2]^{-1}$. The NIS technique is shown in Table 3.1.

We may construct an approximation of the target pdf by means of a discrete random measure $\bar{\pi}^M$ using the TIWs and the set of samples as

$$\bar{\pi}^M(d\boldsymbol{\theta}) = \sum_{i=1}^M \bar{w}^{(i)} \delta_{\boldsymbol{\theta}^{(i)}}(d\boldsymbol{\theta}).$$

Integrals w.r.t. this random measure can be obtained, in turn, as

$$(f, \bar{\pi}^M) = \sum_{i=1}^M \bar{w}^{(i)} f(\boldsymbol{\theta}^{(i)}).$$

Table 3.1: Nonlinear importance sampling (NIS) with target $\pi(\boldsymbol{\theta})$.

1. Draw a set of M samples $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{M}$ from the proposal pdf $q(\boldsymbol{\theta})$.
2. Compute the unnormalized IWs as $w^{(i)*} \propto \pi(\boldsymbol{\theta}^{(i)})/q(\boldsymbol{\theta}^{(i)})$, $i = 1, \ldots, M$.
3. Compute normalized TIWs as

$$\bar{w}^{(i)*} = \varphi^M(w^{(i)*}), \quad \bar{w}^{(i)} = \frac{\bar{w}^{(i)*}}{\sum_{j=1}^{M} \bar{w}^{(j)*}}, \quad i = 1, \ldots, M.$$

3.2.1 Selecting the transformation of the IWs

The nonlinearity φ^M may be constructed in multiple ways. In this section we describe and intuitively justify two specific functions based on the "tempering" and the "clipping", respectively, of the standard IWs.

Tempering

In this case, the unnormalized TIWs are obtained as

$$\bar{w}^{(i)*} = \varphi^M(w^{(i)*}) = (w^{(i)*})^{\gamma}, \quad i = 1, \ldots, M,$$

where $0 < \gamma \leq 1$ is a tempering parameter. This transformation reduces the variations of the IWs, yielding more evenly distributed TIWs and increasing the ESS. While in simple examples this procedure provides a remarkable reduction of the weight variations and an increase of the ESS, we have found that in complex problems it is often not enough to prevent degeneracy.

In this particular case, the TIW $\bar{w}^{(i)}$ associated to a sample $\boldsymbol{\theta}^{(i)}$ only depends on the corresponding standard IW $w^{(i)}$ and the tempering coefficient, but not on the whole set of IWs.

Clipping

Another simple transformation is the clipping, or truncation, of the $M_T < M$ highest IWs. Since the highest weights $w^{(i)}$ usually correspond to the most representative samples $\boldsymbol{\theta}^{(i)}$, we thus obtain M_T flat (non-negligible) TIWs in the region of interest of $\boldsymbol{\theta}$.

To be specific, consider a permutation $i_1, \ldots, i_M$ of the indices in $\{1, ..., M\}$ such that $w^{(i_1)*} \geq \ldots \geq w^{(i_M)*}$ and choose $M_T < M$. We select a

threshold value $\mathcal{T}^M = w^{(i_{M_T})*}$ and apply clipping to the IWs $w^{(i_k)*} \geq \mathcal{T}^M$, $k = 1, \ldots, M_T - 1$. Thus, the unnormalized TIWs $\bar{w}^{(i)*}$, $i = 1, \ldots, M$, are computed from the original IWs $w^{(i)*}$ as[1]

$$\bar{w}^{(i)*} = \varphi^M(w^{(i)*}) = \min(w^{(i)*}, \mathcal{T}^M). \tag{3.1}$$

Note that, since $\mathcal{T}^M = w^{(i_{M_T}*)}$, the number of samples with equal TIWs is exactly M_T.

A bit surprisingly, the selection of the parameter M_T in relation to the total number of samples M is not crucial. Indeed, M_T should be simply large enough to "identify" the region where the posterior probability mass is located. In practice, we have found that choosing $M_T/M = 0.1$ works well for many examples. In a more general setup, the analysis presented in Chapter 4 shows that the approximation of integrals using the NIS scheme converges with rate proportional to $1/\sqrt{M}$, as long as $M_T \leq \sqrt{M}$.

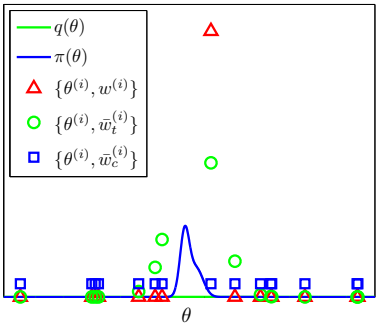

Figure 3.1: Illustration of NIS with tempered, $\bar{w}_t^{(i)}$, and clipped, $\bar{w}_c^{(i)}$, TIWs.

In Figure 3.1 we illustrate the NIS approach with a tempering and a clipping transformation. In this case the target pdf is very narrow w.r.t. the proposal pdf, and the standard IWs $w^{(i)}$ yield an ESS close to 1. The tempered TIWs $\bar{w}_t^{(i)}$ are more evenly distributed, resulting in an increased ESS. Finally, the best M_T samples are assigned equal clipped TIWs $\bar{w}_c^{(i)}$, which yields a predefined ESS and allows to identify the region of high probability of the target pdf. This is particularly useful when implemented in an adaptive manner, since identifying the best set of samples at each iteration allows to improve the proposal pdf for the next iteration.

[1] According to equation (3.1) and the definition of the threshold $\mathcal{T}^M$, φ^M is a function of both the complete weight set $\{w^{(j)*}\}_{j=1}^M$ and the index of the weight to be transformed, i.e., $\varphi^M : \{w^{(j)*}, j = 1, \ldots, M\} \times \{1, \ldots, M\} \to [1, +\infty)$.

3.3 Nonlinear population Monte Carlo

In this section we introduce the nonlinear PMC (NPMC) algorithm, which is an iterative version of the NIS technique and the core contribution of this thesis. We adopt a simple proposal update scheme, where the importance functions are multivariate Gaussian pdfs with moments matched to the latest approximation of the posterior distribution. Besides the basic version of the algorithm, we propose an adaptive version where the transformation of the IWs is only applied when the value of the ESS is below a certain threshold.

We aim to approximate iteratively the target pdf $\pi(\boldsymbol{\theta})$. For simplicity, we select the importance functions in the PMC scheme as multidimensional Gaussian densities. The initial proposal distribution is selected arbitrarily (in the Bayesian approach it can be selected as the prior, i.e., $q_1(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$). In the subsequent iterations we construct the proposal distribution as a multivariate Gaussian distribution

$$q_\ell(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\ell-1}, \boldsymbol{\Sigma}_{\ell-1}), \quad \ell = 2, \ldots, L,$$

where $\boldsymbol{\mu}_{\ell-1}$ is the mean vector and $\boldsymbol{\Sigma}_{\ell-1}$ is a positive definite covariance matrix. These parameters are chosen to match the moments of the distribution described by the discrete measure obtained at the previous iteration. In particular, we compute the mean and covariance at each iteration ℓ as

$$\boldsymbol{\mu}_\ell = \sum_{i=1}^{M} \bar{w}_\ell^{(i)} \boldsymbol{\theta}_\ell^{(i)} \quad \text{and} \tag{3.2}$$

$$\boldsymbol{\Sigma}_\ell = \sum_{i=1}^{M} \bar{w}_\ell^{(i)} (\boldsymbol{\theta}_\ell^{(i)} - \boldsymbol{\mu}_\ell)(\boldsymbol{\theta}_\ell^{(i)} - \boldsymbol{\mu}_\ell)^\top, \tag{3.3}$$

where $\{\boldsymbol{\theta}_\ell^{(i)}, \bar{w}_\ell^{(i)}\}_{i=1}^M$ is the set of samples and TIWs available after the ℓ-th iteration. Note that this particular proposal update scheme is not a constraint of the algorithm. The importance functions can be designed as freely as in the standard PMC method.

The key modification of the algorithm is the computation of TIWs. We introduce a sequence of nonlinear, real positive functions φ_ℓ^M, $\ell = 1, \ldots, L$, which depend both on the iteration index ℓ and the size-M sample at the ℓ-th iteration. The unnormalized TIWs are computed as $\bar{w}_\ell^{(i)*} = \varphi_\ell^M(w_\ell^{(i)*})$, $i = 1, \ldots, M$, where $w_\ell^{(i)*}$ is the standard unnormalized IW associated to the sample $\boldsymbol{\theta}_\ell^{(i)}$. The proposed generic algorithm is outlined in Table 3.2.

Table 3.2: Nonlinear PMC with target $\pi(\boldsymbol{\theta})$.

Iteration ($\ell = 1, \ldots, L$)**:**

1. Select the proposal pdf $q_\ell(\boldsymbol{\theta})$:
   - At iteration $\ell = 1$ select the proposal $q_1(\boldsymbol{\theta})$ arbitrarily.
   - At iterations $\ell = 2, \ldots, L$ the proposal $q_\ell(\boldsymbol{\theta})$ is the Gaussian approximation of $\pi(\boldsymbol{\theta})$ obtained at iteration $\ell - 1$.
2. Draw a set of M independent samples $\Theta_\ell^M = \{\boldsymbol{\theta}_\ell^{(i)}\}_{i=1}^M$ from $q_\ell(\boldsymbol{\theta})$.
3. Compute the unnormalized IWs
$$w_\ell^{(i)*} \propto \frac{\pi(\boldsymbol{\theta}_\ell^{(i)})}{q_\ell(\boldsymbol{\theta}_\ell^{(i)})}, \quad i = 1, \ldots, M. \tag{3.4}$$
4. Compute normalized TIWs as
$$\bar{w}_\ell^{(i)*} = \varphi_\ell^M(w_\ell^{(i)*}), \quad \bar{w}_\ell^{(i)} = \frac{\bar{w}_\ell^{(i)*}}{\sum_{j=1}^M \bar{w}_\ell^{(j)*}}, \quad i = 1, \ldots, M. \tag{3.5}$$
5. Construct a Gaussian approximation $q_{\ell+1}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)$ of the target $\pi(\boldsymbol{\theta})$, where the mean vector and covariance matrix are computed as in equations (3.2) and (3.3).

If the nonlinear transformation is of tempering type, the TIWs are computed at each iteration, $\ell = 1, \ldots, L$, as

$$\bar{w}_\ell^{(i)*} = \varphi_\ell^M(w_\ell^{(i)*}) = (w_\ell^{(i)*})^{\gamma_\ell}, \quad i = 1, \ldots, M,$$

where $0 < \gamma_\ell \leq 1$. The sequence γ_ℓ, $\ell = 1, \ldots, L$, has to be adapted along the iterations, taking low values at the first steps and getting closer to 1 as the algorithm converges. The sequence γ_ℓ can be selected a priori, regardless of the values of the IWs. For instance, it may be constructed as a polynomial function $\gamma_\ell \propto \ell^m$, $m \in \mathbb{N}$, or a sigmoid function $\gamma_\ell = \frac{1}{1+e^{-\ell}}$ of the iteration index ℓ.

On the contrary, if we consider a clipping transformation of the IWs, the

unnormalized TIWs are computed at each iteration $\ell = 1, \ldots, L$ as

$$\bar{w}_\ell^{(i)*} = \varphi_\ell^M(w_\ell^{(i)*}) = \min(w_\ell^{(i)*}, \mathcal{T}_\ell^M), \tag{3.6}$$

where $\mathcal{T}_\ell^M = w_\ell^{(i_{M_T}*)}$ corresponds to the M_T-th highest unnormalized IW at iteration ℓ. The clipping transformation of the IWs ensures a baseline ESS of M_T at all iterations, which allows for a robust update of the proposal distribution for the next iteration.

At each iteration $\ell = 1, \ldots, L$, we obtain a discrete approximation of the target distribution with density $\pi(\boldsymbol{\theta})$

$$\bar{\pi}_\ell^M(d\boldsymbol{\theta}) = \sum_{i=1}^{M} \bar{w}_\ell^{(i)} \delta_{\boldsymbol{\theta}_\ell^{(i)}}(d\boldsymbol{\theta}) \tag{3.7}$$

and integrals of the form (f, π) can be approximated, in turn, as

$$(f, \bar{\pi}_\ell^M) = \sum_{i=1}^{M} \bar{w}_\ell^{(i)} f(\boldsymbol{\theta}_\ell^{(i)}). \tag{3.8}$$

The NESS obtained with TIWs can be approximated at each iteration, $\ell = 1, \ldots, L$, as

$$\bar{M}_\ell^{neff} = \frac{1}{M \sum_{i=1}^{M} (\bar{w}_\ell^{(i)})^2}. \tag{3.9}$$

A resampling step can be performed at each iteration in order to obtain unweighted samples if required. However, is not necessary for our choice of proposal update and it would introduce an additional Monte Carlo error, increasing the variance of the resulting estimates. As already discussed in Chapter 2, SMC methods often introduce a resampling step to overcome the "slight" degeneracy arising in dynamical settings, where the observations are collected sequentially in small sets and the posterior distribution changes slowly over time. However, in static problems of the kind addressed here, all observations are available in a batch beforehand, which yields an extremely peaky posterior distribution, specially in high-dimensional spaces. For this reason, the degeneracy of the weights can be severe in this case, and a plain resampling step is by no means the technique to avoid it. However, the nonlinear transformation of the IWs and, in particular, the clipping technique, avoids the degeneracy of the IWs in a simple and efficient way. It can be applied independently of the proposal distribution and it only requires the setting of the clipping parameter M_T, which can be selected as $M_T \leq \sqrt{M}$. Similarly to the resampling step in sequential settings, the use

of TIWs introduces an additional approximation error (to be analyzed in Chapter 4). However, it avoids numerical problems in the proposal update and renders the approximation to the target distribution much more stable.

The basic NPMC algorithm proposed here is mainly suitable for approximating unimodal and light-tailed target distributions, which can be reasonably modeled by a Gaussian distribution. In this case, the proposal pdf is expected to "approach" the target pdf along the iterations, as the algorithm converges, yielding an increasing ESS. On the contrary, if the target presents multimodality or heavy-tails, the algorithm can yield approximations of the target distribution, and integrals (f, π), but the proposal will not converge to the target distribution.

3.3.1 Modified NPMC

The nonlinear transformation φ_ℓ^M is most useful at the first iterations of the NPMC algorithm, when the proposal density is generally much broader than the target density and the standard IWs may display high variability. In fact, in some applications it may be possible to remove the nonlinear transformation after a few iterations, when the proposal is closer to the target.

Thus, we propose a modification of the NPMC algorithm which consists in applying the nonlinear transformation only if the ESS M_ℓ^{eff} computed from the standard normalized IWs $w_\ell^{(i)}$ is below a specified threshold M_{min}^{eff}. We recommend that the threshold M_{min}^{eff} be a relatively large value (e.g., $\frac{M}{2} \leq M_{min}^{eff} < M$) to ensure that the algorithm is sufficiently stable before removing the transformation. The modified algorithm only differs from the NPMC in step 4, which is outlined in Table 3.3.

Table 3.3: Modified NPMC algorithm.

Step 4 of the NPMC algorithm is replaced by the following computations:

4. Compute the normalized IWs $w_\ell^{(i)} = w_\ell^{(i)*} / \sum_{j=1}^M w_\ell^{(j)*}$ and the ESS $M_\ell^{eff} = [\sum_{i=1}^M (w_\ell^{(i)})^2]^{-1}$.

   If $M_\ell^{eff} < M_{min}^{eff}$, compute normalized TIWs $\bar{w}_\ell^{(i)*} = \varphi_\ell^M(w_\ell^{(i)*})$, $\bar{w}_\ell^{(i)} = \bar{w}_\ell^{(i)*} / \sum_{j=1}^M \bar{w}_\ell^{(j)*}$, $i = 1, \ldots, M$. Otherwise, set $\bar{w}_\ell^{(i)} = w_\ell^{(i)}$.

3.4 Adaptive nonlinear mixture PMC

The original MPMC algorithm, proposed in [30] and described in Section 2.6.3, provides adaptation rules for the approximation of a multimodal target pdf by means of a Gaussian mixture model. This algorithm is very flexible and allows the modelling of a broad variety of target distributions. However, the same as most IS and PMC methods, it suffers from severe degeneracy of the IWs when either the dimension K or the number of observations N is high. Additionally, as it requires the adaptation of parameters of multiple Gaussian components with a limited number of samples, this method is particularly sensitive to the curse of dimensionality. Thus, a very large number of samples is required for this method to provide good results. In spite of these limitations, the MPMC algorithm has been used in practice for the estimation of cosmological parameters, and preferred to existing MCMC alternatives [91, 90, 164].

In this section we introduce a modification of the MPMC algorithm, termed adaptive nonlinear MPMC (NMPMC). The proposed algorithm performs nonlinear transformations to the IWs in order to mitigate the weight degeneracy phenomenon. Thus, in the NMPMC algorithm we additionally compute TIWs $\bar{w}_\ell^{(i)}$ as in equation (3.5), which we use to update the mixture component parameters. We also incorporate an adaptation mechanism for the number of mixture components, allowing to efficiently approximate arbitrary multimodal target distributions in high-dimensional parameter spaces.

3.4.1 Adaptation of the number of components

The original MPMC algorithm assumes a fixed number of components D (which needs to be overestimated in general), hence the final outcome of the algorithm does not provide any information about the number of components required to adequately approximate a target pdf $\pi(\boldsymbol{\theta})$. In this work we propose an extension of the MPMC which incorporates an update step of the number of components D, along the iterations. We consider an initial number of components D_1 and perform pruning and merging operations to the mixture components, reducing D_ℓ over the iterations.

The pruning operation consists in removing the d-th mixture component when its associated weight falls below a prescribed threshold μ_{prn}, i.e., $\alpha_{\ell+1,d} < \mu_{prn}$, as suggested in [164]. The merging operation allows to fuse two similar mixture components $q_{\ell+1,i}$ and $q_{\ell+1,j}$ when the distance $\mathcal{D}_{i,j} = \mathcal{D}(q_{\ell+1,i}||q_{\ell+1,j}) + \mathcal{D}(q_{\ell+1,j}||q_{\ell+1,i})$ is less than a second threshold

μ_{mrg}. The parameters of the resulting component are obtained as the average of the parameters of the original components. The KLD can be computed exactly in the case of Gaussian mixtures, and can be approximated by exact Monte Carlo sampling in the case of t mixtures. Up to one merging and any number of pruning operations are performed at each iteration of the algorithm. The thresholds μ_{prn} and μ_{mrg} are set a priori. The proposed adaptive NMPMC algorithm is outlined in Table 3.4.

Table 3.4: Adaptive nonlinear MPMC algorithm.

Iteration ($\ell = 1, \ldots, L$)**:**

1. Generate a sample $\{\boldsymbol{\theta}_\ell^{(i)}\}_{i=1}^M$ from the current mixture proposal $q_\ell(\boldsymbol{\theta})$ in equation (2.28) with $D = D_\ell$ components.

2. For $i = 1, \ldots, M$, compute IWs $w_\ell^{(i)*}$ as in equation (3.4) and mixture posterior probabilities $\rho_{\ell,d}^{(i)}$ as in equation (2.29), with $D = D_\ell$.

3. For $i = 1, \ldots, M$, compute TIWs $\bar{w}_\ell^{(i)}$ as in equation (3.5).

4. Update the component weights $\alpha_{\ell+1,d}$, $d = 1, ..., D_\ell$, and parameters $\boldsymbol{\beta}_{\ell+1,d}$ of each component according to equations (2.30) and (2.31), respectively, but using TIWs $\bar{w}_\ell^{(i)}$ instead of standard IWs $w_\ell^{(i)}$.

5. Set $\tilde{D} = D_\ell$. Compute the distance $\mathcal{D}_{i,j}$ between each pair of mixture components $q_{\ell+1,i}$ and $q_{\ell+1,j}$, for $i, j = 1, \ldots, D_\ell$.

   If $\mathcal{D}_{i,j} < \mu_{mrg}$, merge components i and j. The overall weight is computed as $\alpha_{\ell+1,i} = \alpha_{\ell+1,i} + \alpha_{\ell+1,j}$ and the parameters as $\boldsymbol{\mu}_{\ell+1,i} = (\boldsymbol{\mu}_{\ell+1,i} + \boldsymbol{\mu}_{\ell+1,j})/2$ and $\boldsymbol{\Sigma}_{\ell+1,i} = (\boldsymbol{\Sigma}_{\ell+1,i} + \boldsymbol{\Sigma}_{\ell+1,j})/2$. Remove the j-th component setting $\alpha_{\ell+1,j} = 0$ and $\tilde{D} = \tilde{D} - 1$.

6. For $i = 1, \ldots, \tilde{D}$, if $\alpha_{\ell+1,i} < \mu_{prn}$, remove the i-th component setting $\alpha_{\ell+1,i} = 0$ and $\alpha_{\ell+1,j} = \alpha_{\ell+1,j} / \sum_{k=1}^{\tilde{D}} \alpha_{\ell+1,k}$, $j = 1, \ldots, \tilde{D}$.

7. Update $\tilde{D}$ according to the number of pruned components and set $D_{\ell+1} = \tilde{D}$.

This adaptive NMPMC scheme provides valuable information about the number of components required to represent the target pdf and can also

alleviate the computational demands of the algorithm (as it is simpler to draw samples from mixtures with less components). Opposite to alternative MCMC methods, this algorithm is robust against the "you've only seen where you've been" phenomenon [147], since it explores all the space of $\boldsymbol{\theta}$ described by the prior distribution.

3.5 Particle NPMC for state-space models

In this section we present an NPMC method, termed particle NPMC (PNPMC), for the approximation of posterior distributions arising in state-space models of the type described in Section 2.2.2. First, we tackle the problem where only the marginal posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$ of the unknown parameters is required, and we propose a PNPMC method that resorts to a PF approximation of the likelihood function, similarly to the PMCMC method. Then, we describe a straightforward extension that allows to approximate offline the joint posterior distribution of the parameters and the hidden state $p(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$. In this section we only consider the clipping transformation of the IWs because it yields the best results in practice. However, alternative nonlinear transformations can also be applied.

3.5.1 Particle NPMC targeting $p(\boldsymbol{\theta}|\mathbf{y})$

We first consider as a target density the marginal posterior pdf of the parameters given the observation vector $\mathbf{y}$, i.e., $\pi(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y})$. As already discussed in Section 2.2.2, the likelihood function $p(\mathbf{y}|\boldsymbol{\theta})$ given by equation (2.9) cannot be evaluated exactly in general. However, we can resort to a standard PF to compute approximations to this likelihood by means of equations (2.17) and (2.19). Thus, in similar vein as in the PMCMC algorithm, the densities $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ and $p(\mathbf{y}|\boldsymbol{\theta})$ required in steps 2 and 3 of the PNPMC algorithm are replaced by their particle approximations, which are computed via a standard PF as described in Table 2.1. The PNPMC algorithm for the approximation of $p(\boldsymbol{\theta}|\mathbf{y})$ in state-space models is displayed in Table 3.5.

As in the general NPMC algorithm described in Section 3.3, the choice of a Gaussian approximation of the proposal $q_{\ell+1}(\boldsymbol{\theta})$ in step 5 is arbitrary (and done for simplicity here). Any other family of pdfs can be used without modifying the rest of the algorithm. Indeed, if the target pdf $p(\boldsymbol{\theta}|\mathbf{y})$ is multimodal, the proposed PNPMC algorithm can be straightforwardly combined with a mixture proposal pdf as in the MPMC method of [30].

Table 3.5: Particle NPMC targeting $p(\boldsymbol{\theta}|\mathbf{y})$ in a state-space model.

Iteration ($\ell = 1, \ldots, L$)**:**

1. Draw a set of M samples $\{\boldsymbol{\theta}_\ell^{(i)}\}_{i=1}^M$ from the proposal density $q_\ell(\boldsymbol{\theta})$:
   - at iteration $\ell = 1$, let $q_1(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$.
   - at iterations $\ell = 2, \ldots, L$ the proposal $q_\ell(\boldsymbol{\theta})$ is the Gaussian approximation of $p(\boldsymbol{\theta}|\mathbf{y})$ obtained at iteration $\ell - 1$.
2. For $i = 1, \ldots, M$, run a PF with J particles targeting $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}_\ell^{(i)})$ and compute the marginal likelihood estimate $\hat{p}_\ell^J(\mathbf{y}|\boldsymbol{\theta}_\ell^{(i)})$.
3. Compute the unnormalized IWs as
$$w_\ell^{(i)*} \propto \frac{\hat{p}_\ell^J(\mathbf{y}|\boldsymbol{\theta}_\ell^{(i)})p(\boldsymbol{\theta}_\ell^{(i)})}{q_\ell(\boldsymbol{\theta}_\ell^{(i)})}, \quad i = 1, \ldots, M.$$
4. Compute normalized TIWs, $\bar{w}_\ell^{(i)}$, by *clipping* the original IWs as
$$\bar{w}_\ell^{(i)*} = \min(w_\ell^{(i)*}, \mathcal{T}_\ell^{M_T}), \quad \bar{w}_\ell^{(i)} = \bar{w}_\ell^{(i)*} / \sum_{j=1}^M \bar{w}_\ell^{(j)*}, \qquad i = 1, \ldots, M$$
   where the threshold value $\mathcal{T}_\ell^{M_T}$ denotes the M_T-th highest unnormalized IW $w_\ell^{(i)*}$, with $1 < M_T < M$.
5. Construct a Gaussian approximation $q_{\ell+1}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)$ of the posterior $p(\boldsymbol{\theta}|\mathbf{y})$, where the mean vector and covariance matrix are computed as in equations (3.2) and (3.3), respectively.

At each iteration of the PNPMC algorithm we can construct a discrete approximation of the posterior pdf $p(\boldsymbol{\theta}|\mathbf{y})$, based on the set of samples and TIWs, as in equation (3.7) and integrals w.r.t. to it can be approximated as in equation (3.8).

3.5.2 Particle NPMC targeting $p(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$

The PNPMC method proposed in Section 3.5.1 may be readily applied to the approximation of the full joint posterior $p(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$, in an manner equivalent to the PMCMC algorithm. We consider a sampling mechanism of the form $q(\boldsymbol{\theta})\hat{p}^J(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$, where samples $\boldsymbol{\theta}^{(i)}$ are again generated from the latest proposal $q(\boldsymbol{\theta})$ and $\mathbf{x}^{(i)}$ are drawn form the approximation $\hat{p}^J(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}^{(i)})$ obtained my means of a PF (the iteration index has been omitted for simplicity). Then, the standard, unnormalized IW associated to the pair $(\boldsymbol{\theta}^{(i)}, \mathbf{x}^{(i)})$ is computed as

$$\begin{aligned} w^{(i)*} &= \frac{\hat{p}^J(\boldsymbol{\theta}^{(i)}, \mathbf{x}^{(i)}|\mathbf{y})}{q(\boldsymbol{\theta}^{(i)})\hat{p}^J(\mathbf{x}^{(i)}|\mathbf{y}, \boldsymbol{\theta}^{(i)})} \\ &\propto \frac{\hat{p}^J(\mathbf{x}^{(i)}, \mathbf{y}|\boldsymbol{\theta}^{(i)})p(\boldsymbol{\theta}^{(i)})}{q(\boldsymbol{\theta}^{(i)})\hat{p}^J(\mathbf{x}^{(i)}|\mathbf{y}, \boldsymbol{\theta}^{(i)})} \propto \frac{\hat{p}^J(\mathbf{y}|\boldsymbol{\theta}^{(i)})p(\boldsymbol{\theta}^{(i)})}{q(\boldsymbol{\theta}^{(i)})} \end{aligned}$$

and is independent of $\mathbf{x}$. At each iteration, $\ell = 1, \ldots, L$, the algorithm yields a discrete approximation of the posterior distribution of the parameters $\boldsymbol{\theta}$ and the unobserved populations $\mathbf{x}$ constructed as

$$\bar{p}_\ell^{M,J}(d\boldsymbol{\theta}|\mathbf{y}) = \sum_{i=1}^{M} \bar{w}_\ell^{(i)} \delta_{\boldsymbol{\theta}_\ell^{(i)}}(d\boldsymbol{\theta}) \quad \text{and} \quad \bar{p}_\ell^{M,J}(d\mathbf{x}|\mathbf{y}) = \sum_{i=1}^{M} \bar{w}_\ell^{(i)} \delta_{\mathbf{x}_\ell^{(i)}}(d\mathbf{x}),$$

respectively.

The PNPMC method processes a set of M i.i.d. samples at each iteration, requiring a low number of iterations (around 10 for the type of problems addressed here) for convergence to the target distribution. The bulk of the computational cost of PNPMC, as well as of PMCMC, is the particle approximation of the likelihood function. In the PMCMC algorithm the samples $\boldsymbol{\theta}^{(i)}$ are processed sequentially (one after the other), and this process cannot be parallelized. On the contrary, at each iteration ℓ of the PNPMC method, the process of drawing M samples $\boldsymbol{\theta}_\ell^{(i)}$ and computing the associated IWs $w_\ell^{(i)*}$ can be performed independently for each sample i. Thus, steps 1, 2 and 3 of the PNPMC algorithm can be easily parallelized, reducing the total execution time up to that of a single sample $\boldsymbol{\theta}_\ell^{(i)}$. On

the other hand, steps 4 and 5 require the complete set of samples and weights $\{\boldsymbol{\theta}_\ell^{(i)}, w_\ell^{(i)*}\}_{i=1}^M$ and must be performed in a centralized manner. However, these computations often have a negligible cost in comparison with the likelihood approximation. Thus, the parallelization of the PNPMC method can allow for a reduction in execution time up to a factor $\approx 1/M$. Note that we refer here to the parallelization of the NPMC method and not of the PF used to approximate the likelihood function.

3.6 Connections with other methods

The nonlinear PMC methods proposed here present similarities with a set of related techniques. In this section we review some of these connections.

3.6.1 Tempering techniques

The main idea behind the NIS and NPMC schemes is to smooth the IWs to increase the efficiency of the IS technique and allow for a robust proposal update. We propose two distinct ways of smoothing the IWs: tempering and clipping. The tempering transformation of the IWs is closely related to the *simulated tempering* of the target density, which has been widely studied in the MCMC literature [74, 121].

The necessity of smoothing the target distribution in order to facilitate the adaptation from a wide prior distribution with an affordable computational cost has been addressed many times in the literature. For example, the AIS algorithm of [130] introduces a sequence of tempered target distributions, with the assumption that if each two consecutive functions are close, the adaptation can be performed in a robust way. However, this method does not guarantee that this actually happens, and additionally it requires the selection of a number of transition kernels and associated auxiliary target distributions. On the other hand, the IBIS method of [41] also attempts to induce a tempering effect on the intermediate posterior distributions $p(\boldsymbol{\theta}|\mathbf{y}_{0:n_\ell})$ by incorporating the observations in a sequential manner. However, this procedure requires the setting of the incorporation schedule and does not ensure to avoid degeneracy either.

In [49] a piecewise constant SIS algorithm was proposed, which aims at reducing the cost of traditional PFs by approximating the likelihood with a mixture of uniform distributions over pre-defined cells or bins. The idea behind this algorithm resembles the clipping procedure introduced in this thesis.

Connection with SMC samplers

As noted in Chapter 2, tempering techniques have been specifically considered within the SMC framework of [47, 84, 20]. However, the IWs in the SMC methodology of [47] are computed in the conventional manner, and tempering is only applied to the target density [20]. Therefore, these methods depart from the NPMC scheme, as the same set of samples in the parameter space (even drawn from the same proposal) would be weighted differently. However, it is possible to derive a NPMC algorithm with tempering within the framework of [47], as shown here, under some constraints on the choice of the importance functions. Unfortunately, the latter constraints rule out the class of proposals $q_\ell(\boldsymbol{\theta})$ introduced in Section 3.3, where each function $q_\ell(\boldsymbol{\theta})$ is selected by matching the empirical moments of the population $\{\boldsymbol{\theta}_{\ell-1}^{(i)}, w_{\ell-1}^{(i)}\}_{i=1}^M$.

We address the question of whether a NPMC algorithm with tempering can be obtained as a particular case of the SMC sampler of [47] by a proper choice of the backward and forward kernels. The answer is partially positive. Indeed, consider the generic weight function in equation (2.24). If we select a sequence of exponents $0 < \gamma_1 < \gamma_2 < \ldots < \gamma_L = 1$ and define $\pi_\ell(\boldsymbol{\theta}_\ell) = \pi(\boldsymbol{\theta}_\ell)^{\gamma_\ell}$, then we can equate

$$\text{IW} \equiv \frac{\pi(\boldsymbol{\theta}_\ell)^{\gamma_\ell} b_{\ell-1}(\boldsymbol{\theta}_{\ell-1}|\boldsymbol{\theta}_\ell)}{\pi(\boldsymbol{\theta}_{\ell-1})^{\gamma_{\ell-1}} q_\ell(\boldsymbol{\theta}_\ell|\boldsymbol{\theta}_{\ell-1})} = \frac{\pi(\boldsymbol{\theta}_\ell)^{\gamma_\ell}}{q_\ell(\boldsymbol{\theta}_\ell|\boldsymbol{\theta}_{\ell-1})^{\gamma_\ell}} \equiv \text{TIW},$$

and solve for the backward kernel density, namely

$$b_{\ell-1}(\boldsymbol{\theta}_{\ell-1}|\boldsymbol{\theta}_\ell) \propto \pi(\boldsymbol{\theta}_{\ell-1})^{\gamma_{\ell-1}} q_\ell(\boldsymbol{\theta}_\ell|\boldsymbol{\theta}_{\ell-1})^{1-\gamma_\ell}. \tag{3.10}$$

However, it is not possible to make equation (3.10) hold for *any* proposal scheme and, in particular, it cannot hold for the type of proposals introduced in Section 3.3. To be precise, the backward kernel density $b_\ell(\boldsymbol{\theta}_{\ell-1}|\boldsymbol{\theta}_\ell)$ can be chosen as in equation (3.10) if the i-th sample in the ℓ-th iteration is drawn conditional on i-th sample from the iteration $\ell - 1$. This is the usual case, e.g., in particle filtering applications where the variables of interest are dynamic and a forward kernel density is actually part of the model. If $q_\ell(\boldsymbol{\theta}_\ell|\boldsymbol{\theta}_{\ell-1}) = q_\ell(\boldsymbol{\theta}_\ell)$ is designed simply from the statistics of the weighted population $\{\boldsymbol{\theta}_{\ell-1}^{(i)}, w_{\ell-1}^{(i)}\}_{i=1}^M$, then the backward kernel becomes independent of the forward kernel, i.e.,

$$b_{\ell-1}(\boldsymbol{\theta}_{\ell-1}|\boldsymbol{\theta}_\ell) \propto \pi(\boldsymbol{\theta}_{\ell-1})^{\gamma_{\ell-1}} q_\ell(\boldsymbol{\theta}_\ell)^{1-\gamma_\ell} \propto \pi(\boldsymbol{\theta}_{\ell-1})^{\gamma_{\ell-1}}$$

and the weight function of the NPMC algorithm with tempering cannot be reproduced.

3.6.2 MCMC methods

As already discussed in Chapter 2, PMC and MCMC methods share common fields of application and present some common features. Both techniques require the selection of a proposal pdf and rely on the computation of the ratio $\pi(\boldsymbol{\theta})/q(\boldsymbol{\theta})$ for the computation of the IWs and the acceptance rate, respectively. Even though PMC and MCMC algorithms often require very similar computations for each sample $\boldsymbol{\theta}^{(i)}$, PMC and the proposed NPMC methods have a set of relevant advantages w.r.t. their MCMC counterparts:

- PMC methods provide independent sets of samples at all iterations and do not require a burn-in period.
- PMC methods allow for a much simpler parallel implementation than MCMC methods.
- Contrary to MCMC algorithms, that require a careful choice of the proposal tuning parameter, the proposed NPMC methods do not require the accurate fitting of any parameters.

On the other hand, the nonlinearity applied in the NPMC scheme mitigates weight degeneracy, which is the main problem arising in conventional IS-based methods, dramatically increasing its efficiency in high-dimensional spaces. However, no solution has been provided so far for the discussed problems of MCMC methods and they are often applied at a very high computational cost. As a consequence, we claim that the total number of samples, ML required by the NPMC methods can be significantly lower than that of MCMC, I.

Al already noted in Section 3.5, the PNPMC method for state-space models is closely related to the PMCMC algorithm of [6], since both methods rely on a particle approximation of the likelihood function and have equivalent computational cost for each pair of samples $\{\boldsymbol{\theta}, \mathbf{x}\}$.

3.7 Summary

In this chapter we have described the proposed Monte Carlo algorithms for simulating from complex target distributions in high dimension and approximating integrals w.r.t. them. In particular, we have introduced a novel family of PMC algorithms that addresses the main weakness of standard IS techniques, which is the degeneracy of the IWs and the subsequent inefficiency in high-dimensional spaces. We have introduced a

nonlinear IS technique, which computes nonlinearly transformed IWs and yields an increased number of effective samples, and discussed some possible choices of the nonlinear transformation function.

We have proposed to use NIS within a PMC algorithm. Even when the proposed technique can be applied independently of the proposal update scheme, we have explored two different and generic proposal distribution choices. The basic NPMC algorithm constructs the importance functions as multivariate Gaussian distributions and it is thus specially suited for approximating unimodal, light-tailed target distributions. As a generalization for arbitrary multimodal distributions, we have applied the NPMC method to extend the MPMC algorithm of [30]. The proposed method introduces nonlinear transformations of the IWs to mitigate the degeneracy problem and incorporates an adaptation mechanism for the number of mixture components. The resulting adaptive NMPMC algorithm provides a very general and flexible tool to efficiently approximate arbitrary distributions $\pi(\boldsymbol{\theta})$ in high-dimensional parameter spaces. We have also proposed a particularization of the NPMC method to the problem of offline Bayesian inference in state-space models, where the interest is on the posterior distribution of a set of parameters $\boldsymbol{\theta}$ and hidden states $\mathbf{x}$, given a set of observations $\mathbf{y}$. The proposed algorithm relies on a likelihood approximation computed by means of a PF, and is thus termed particle NPMC.

Chapter 4

Convergence analysis of nonlinear importance sampling

Compared to a standard IS scheme, the nonlinear transformations of the IWs in the NIS method introduce a distortion in the approximating random probability measure, and therefore, it is not apparent, a priori, that this measure should converge in the same way as the measure induced by the standard IWs. Therefore, in this chapter we provide a convergence analysis of the NIS technique. In Section 4.1 we introduce some notation and general assumptions used through the chapter. In Section 4.2 we present an analysis of the error induced by the tempering transformation. In Section 4.3 we introduce asymptotic convergence results for the NIS technique with a clipping transformation, both with exact and approximate IWs. For example, in the kind of problems addressed in Section 2.2.2, the IWs cannot be evaluated exactly but they can, instead, be approximated via particle filtering. Therefore, we look explicitly into the convergence of the approximations of integrals computed using approximate weights (both IWs and TIWs). In particular, we derive explicit convergence rates for the L_2 norms of the approximation errors and show that the approximate weights computed by a standard PF are "good enough" to ensure that these results hold. In Section 4.5 we detail the proofs of the results mentioned above and Section 4.6 is devoted to the conclusions of this chapter.

4.1 Notation and basic assumptions

Let $\pi(\boldsymbol{\theta})$ be the pdf associated to the target probability distribution, let $q(\boldsymbol{\theta})$ be the importance function used to propose samples in an IS scheme (not necessarily normalized) and let $h(\boldsymbol{\theta}) \propto \pi(\boldsymbol{\theta})$ be a function proportional to π, with the proportionality constant independent of $\boldsymbol{\theta}$. The samples drawn from the distribution associated to q are denoted $\boldsymbol{\theta}^{(i)}$, $i = 1, ..., M$, and their associated unnormalized IWs are $w^{(i)*} = h(\boldsymbol{\theta}^{(i)})/q(\boldsymbol{\theta}^{(i)})$, $i = 1, ..., M$.

Let us define the weight function $g(\boldsymbol{\theta}) = h(\boldsymbol{\theta})/q(\boldsymbol{\theta})$ which, in particular, yields $g(\boldsymbol{\theta}^{(i)}) = w^{(i)*}$. We assume that the weight function $g \in B(\mathbb{R}^K)$ is upper bounded, and thus the TIWs satisfy $\bar{w}^{(i)*} \leq \|g\|_\infty = \sup_{\boldsymbol{\theta} \in \mathbb{R}^K} |g(\boldsymbol{\theta})| < \infty$. The support of g is the same as the support of q and π, denoted $\mathsf{S} \subseteq \mathbb{R}^K$. If we assume that both $q(\boldsymbol{\theta}) > 0$ and $\pi(\boldsymbol{\theta}) > 0$ for any $\boldsymbol{\theta} \in \mathsf{S}$, then $g(\boldsymbol{\theta}) > 0$ for every $\boldsymbol{\theta} \in \mathsf{S}$ as well. Also, trivially, $\pi \propto gq$, with the proportionality constant independent of $\boldsymbol{\theta}$. These assumptions are standard for classical IS.

In the sequel we look into the approximation of integrals of the form

$$(f, \pi) = \int I_{\mathsf{S}}(\boldsymbol{\theta}) f(\boldsymbol{\theta}) \pi(\boldsymbol{\theta}) d\boldsymbol{\theta},$$

where $I_{\mathsf{S}}(\boldsymbol{\theta})$ is an indicator function (namely, $I_{\mathsf{S}}(\boldsymbol{\theta}) = 1$ if $\boldsymbol{\theta} \in \mathsf{S}$ and $I_{\mathsf{S}}(\boldsymbol{\theta}) = 0$ otherwise) and f is a bounded real function in the parameter space S. We use $\|f\|_\infty = \sup_{\boldsymbol{\theta} \in \mathsf{S}} |f(\boldsymbol{\theta})| < \infty$ to denote the supremum norm of a bounded function. The set of real bounded functions on S is $B(\mathsf{S}) = \{f : \mathsf{S} \to \mathbb{R} : \|f\|_\infty < \infty\}$.

Assuming that the standard IWs can be computed exactly, the approximation π^M of the target probability measure can be written as

$$\pi^M(d\boldsymbol{\theta}) = \sum_{i=1}^{M} w^{(i)} \delta_{\boldsymbol{\theta}^{(i)}}(d\boldsymbol{\theta}),$$

where $w^{(i)} = \frac{g(\boldsymbol{\theta}^{(i)})}{\sum_{j=1}^{M} g(\boldsymbol{\theta}^{(j)})}$, $i = 1, ..., M$.

The approximation $\bar{\pi}^M$ of the target probability measure generated by the NIS method is constructed from the normalized TIWs $\bar{w}^{(i)}$ as

$$\bar{\pi}^M(d\boldsymbol{\theta}) = \sum_{i=1}^{M} \bar{w}^{(i)} \delta_{\boldsymbol{\theta}^{(i)}}(d\boldsymbol{\theta}),$$

where $\bar{w}^{(i)} = \frac{\varphi^M(g(\boldsymbol{\theta}^{(i)}))}{\sum_{j=1}^{M} \varphi^M(g(\boldsymbol{\theta}^{(j)}))}$, $i = 1, ..., M$. We recall that $\varphi^M : (\mathbb{R}^+)^M \times \{1, \ldots, M\} \to \mathbb{R}^+$ is a transformation function that can depend both on

the complete weight set $\{g(\boldsymbol{\theta}^{(j)})\}_{j=1}^{M}$ and the index i of the weight to be transformed.

The analysis contained in this chapter is concerned with the asymptotic performance of the NIS approximation as the number of samples M grows, but not with the convergence of the NPMC algorithm as the iteration index ℓ increases. Hence, we drop the latter subscript for convenience all through the chapter.

4.2 NIS with tempering

In this section we consider the NIS technique with a transformation of the IWs of the tempering type. In this case, the TIWs can be written as

$$\bar{w}^{(i)} = \frac{g(\boldsymbol{\theta}^{(i)})^{\gamma}}{\sum_{j=1}^{M} g(\boldsymbol{\theta}^{(j)})^{\gamma}}, \quad i = 1, ..., M. \tag{4.1}$$

If $\gamma < 1$ is fixed and $f \in B(\mathsf{S})$ is non-constant, it is apparent that the integral $(f, \bar{\pi}^M)$ does not converge to (f, π) as $M \to \infty$. However, it is straightforward to find an upper bound for the distortion with respect to the conventional IS approximation, (f, π^M), as given by the following proposition.

Proposition 1. *Assume that $g \in B(\mathsf{S})$, $\varphi^M(w) = w^{\gamma}$ and both $0 < \gamma \leq 1$ and $M < \infty$ are fixed. Then, for every $f \in B(\mathsf{S})$,*

$$|(f, \pi^M) - (f, \bar{\pi}^M)| \leq |(f(1 - g^{\gamma-1}), \pi^M)| + \|f\|_{\infty} |(1 - g^{\gamma-1}, \pi^M)|. \tag{4.2}$$

Proof: See Section 4.5.1.

The inequality (4.2) is useful because it yields an upper bound for the distortion $|(f, \pi^M) - (f, \bar{\pi}^M)|$, introduced by the tempering nonlinearity, that depends on the standard IS approximating measure π^M alone. Since $1 - g^{\gamma-1} \in B(\mathsf{S})$, the standard convergence results for IS [65] can be applied to the integrals on the right hand side of (4.2) and, as a consequence,

$$\lim_{M \to \infty} |(f, \pi^M) - (f, \bar{\pi}^M)| \leq |(f(1 - g^{\gamma-1}), \pi)| + \|f\|_{\infty} |(1 - g^{\gamma-1}, \pi)| \tag{4.3}$$

a.s. Moreover, (4.2) also shows that the difference $(f, \pi^M) - (f, \bar{\pi}^M)$ vanishes when $\gamma \to 1$. Indeed, when $\gamma \to 1$, $(1 - g^{\gamma-1}, \pi^M) \to 0$ and $(f(1 - g^{\gamma-1}), \pi^M) \to 0$, hence

$$\lim_{\gamma \to 1} |(f, \pi^M) - (f, \bar{\pi}^M)| = 0.$$

Similarly, from (4.3) we observe that

$$\lim_{\gamma\to 1}\lim_{M\to\infty}|(f,\bar{\pi}^M)-(f,\pi)|=0 \quad \text{a.s.},$$

as intuitively expected.

4.3 NIS with clipping and approximate weights

In this section we look explicitly into the convergence of the estimates of integrals computed using approximate weights. In particular, we provide upper bounds for the estimation errors that hold almost surely (a.s.) and depend explicitly on both the number of generated samples, M, and the approximation error for the IWs.

If the weight function can only be computed approximately, let us denote its approximation as g^ϵ. The resulting random measure is

$$\pi^{M,\epsilon}(d\boldsymbol{\theta})=\sum_{i=1}^{M} w^{(i),\epsilon}\delta_{\boldsymbol{\theta}^{(i)}}(d\boldsymbol{\theta}),$$

where $w^{(i),\epsilon}=\frac{g^\epsilon(\boldsymbol{\theta}^{(i)})}{\sum_{j=1}^{M} g^\epsilon(\boldsymbol{\theta}^{(j)})}$, $i=1,...,M$.

We assume that the nonlinear transformation φ^M of the weights is of a clipping class, as described in Section 3.2.1. We note that, given an index permutation $i_1,\ldots,i_M$ such that $w^{(i_1)*}\geq\ldots\geq w^{(i_M)*}$, the transformation φ^M can be expressed as

$$\varphi^M(w^{(i_k)*})=\begin{cases} w^{(i_{M_T})*}, & \text{for } k=1,\ldots,M_T, \text{ and} \\ w^{(i_k)*}, & \text{for } k=M_T+1,\ldots,M. \end{cases} \tag{4.4}$$

The weighted approximation of $\pi(\boldsymbol{\theta})d\boldsymbol{\theta}$ constructed according to the nonlinear IS scheme is

$$\bar{\pi}^{M,\epsilon}(d\boldsymbol{\theta})=\sum_{i=1}^{M}\bar{w}^{(i),\epsilon}\delta_{\boldsymbol{\theta}^{(i)}}(d\boldsymbol{\theta}),$$

where $\bar{w}^{(i),\epsilon}=\frac{\varphi^M(g^\epsilon(\boldsymbol{\theta}^{(i)}))}{\sum_{j=1}^{M}\varphi^M(g^\epsilon(\boldsymbol{\theta}^{(j)}))}$, $i=1,...,M$.

We make the following assumptions on the weight function, g, and its approximation, g^ϵ.

Assumption 1. *For some $\epsilon \geq 0$, the approximation g^ϵ of the weight function satisfies the inequality*

$$\sup_{\boldsymbol{\theta}\in\mathsf{S}} |g(\boldsymbol{\theta}) - g^\epsilon(\boldsymbol{\theta})| \leq \epsilon \quad a.s.$$

Assumption 2. *The weight function g has a finite upper bound and a positive lower bound. Specifically, there exists a real number $0 < a < \infty$ such that $a^{-1} \leq g(\boldsymbol{\theta}) \leq a$ for every $\boldsymbol{\theta} \in \mathsf{S}$.*

Assumption 3. *The same bounds of g hold for its approximations g^ϵ, $\epsilon \geq 0$. To be specific, $a^{-1} \leq g^\epsilon(\boldsymbol{\theta}) \leq a$ for every $\boldsymbol{\theta} \in \mathsf{S}$ and any $\epsilon \geq 0$.*

Note that if the support set S is compact then Assumption 2 holds whenever $q > 0$ and $h > 0$ in S. Otherwise, the proposal q has to be chosen so that it has heavier tails than π.

The approximations of interest are

$$(f, \pi^{M,\epsilon}) = \sum_{i=1}^{M} f(\boldsymbol{\theta}^{(i)}) w^{(i),\epsilon} \quad \text{and} \quad (f, \bar{\pi}^{M,\epsilon}) = \sum_{i=1}^{M} f(\boldsymbol{\theta}^{(i)}) \bar{w}^{(i),\epsilon}.$$

The following Theorem yields upper bounds for the absolute approximation errors $|(f, \pi^{M,\epsilon}) - (f, \pi)|$ and $|(f, \bar{\pi}^{M,\epsilon}) - (f, \pi)|$ that depend explicitly on M and ϵ.

Theorem 1. *Assume that $M_T \leq \sqrt{M}$ and Assumptions 1, 2 and 3 hold. Then, there exist positive and a.s. finite random variables $W_{f,\upsilon}$ and $\bar{W}_{f,\upsilon}$, independent of M and ϵ, such that*

$$|(f, \pi^{M,\epsilon}) - (f, \pi)| \leq \frac{W_{f,\upsilon}}{M^{\frac{1}{2}-\upsilon}} + C\epsilon \tag{4.5}$$

and

$$|(f, \bar{\pi}^{M,\epsilon}) - (f, \pi)| \leq \frac{\bar{W}_{f,\upsilon}}{M^{\frac{1}{2}-\upsilon}} + C\epsilon \tag{4.6}$$

for every $f \in B(\mathsf{S})$, arbitrarily small $0 < \upsilon < \frac{1}{2}$ and $C < \infty$. Both C and υ are independent of M, M_T and ϵ.

Proof: See Section 4.5.2. □

Theorem 1 yields an upper bound for the (random) absolute error that consists of two terms, one that depends on the number of samples M and another one that depends on the weight approximation error ϵ. As $M \to \infty$, the first term vanishes with the usual Monte Carlo rate of convergence

despite the approximation of the IWs and the clipping transformation. The second term is proportional to the approximation error, hence it only vanishes when the routine used to compute g^ϵ can be made arbitrarily accurate (i.e., $\epsilon \to 0$), typically by increasing the computational effort invested in this calculation.

Remark 1. *In case the weights can be evaluated exactly, it is sufficient to set $\epsilon = 0$ in the inequalities (4.5) and (4.6) to obtain the corresponding upper bounds of the approximation error.*

4.4 NIS with clipping and PF approximation

In this section we introduce a more precise notation for the state-space model (compared to the argument-wise $p(\mathbf{x})$ and $p(\mathbf{y}|\mathbf{x})$ used in the previous chapters), in order to perform the analysis with approximate weights computed using PFs. Assume a discrete-time state space Markov model with state process $\{\mathbf{X}_n\}_{n\geq 0}$ taking values on $\mathcal{X} \subseteq \mathbb{R}^{d_\mathbf{x}}$ and an observation process $\{\mathbf{Y}_n\}_{n\geq 0}$ taking values on $\mathcal{Y} \subseteq \mathbb{R}^{d_\mathbf{y}}$. The prior distribution (probability measure) of the state is now denoted $\tau_0(d\mathbf{x})$ and the transition (Markov) kernel depends on a vector-valued random parameter $\boldsymbol{\Theta}$ that takes values on a compact set $\mathsf{S} \subset \mathbb{R}^K$ and has prior distribution $\mu_0(d\boldsymbol{\theta})$ independent of $\mathbf{X}_0$. In particular, the Markov kernel is now denoted $\tau_{n,\boldsymbol{\theta}}(d\mathbf{x}_n|\mathbf{x}_{n-1})$ and the conditional density of the observations is $u_n(\mathbf{y}_n|\mathbf{x}_n) > 0$ (independent of $\boldsymbol{\theta}$). The latter also yields the likelihood of the signal $\mathbf{x}_n$, hence we often write, for conciseness, $u_n^{\mathbf{y}_n}(\mathbf{x}_n) \triangleq u_n(\mathbf{y}_n|\mathbf{x}_n)$.

At time n, the one-step-ahead predictive distribution of the state $\mathbf{X}_n$ given fixed observations $\mathbf{Y}_{1:n-1} = \mathbf{y}_{1:n-1}$ and a parameter value $\boldsymbol{\Theta} = \boldsymbol{\theta}$ is denoted $\xi_{n,\boldsymbol{\theta}}$, specifically[1], for any Borel subset $A \subset \mathcal{X}$,

$$\xi_{n,\boldsymbol{\theta}}(A) = \mathbb{P}_n\left(\mathbf{X}_n \in A | \mathbf{Y}_{1:n-1} = \mathbf{y}_{1:n-1}, \boldsymbol{\Theta} = \boldsymbol{\theta}\right).$$

The filter measure at time n given observations $\mathbf{Y}_{1:n} = \mathbf{y}_{1:n}$ and parameter $\boldsymbol{\Theta} = \boldsymbol{\theta}$ is denoted $\phi_{n,\boldsymbol{\theta}}$, namely,

$$\phi_{n,\boldsymbol{\theta}}(A) = \mathbb{P}_n\left(\mathbf{X}_n \in A | \mathbf{Y}_{1:n} = \mathbf{y}_{1:n}, \boldsymbol{\Theta} = \boldsymbol{\theta}\right).$$

The predictive measure $\xi_{n,\boldsymbol{\theta}}$ can be expressed in terms of $\tau_{n,\boldsymbol{\theta}}$ and $\phi_{n-1,\boldsymbol{\theta}}$. Specifically, we write $\xi_{n,\boldsymbol{\theta}} = \tau_{n,\boldsymbol{\theta}}\phi_{n-1,\boldsymbol{\theta}}$, meaning that, for any integrable

[1] $\mathbb{P}_n$ denotes the joint probability measure for the set of random variables $\{\mathbf{x}_k\}_{k\leq n} \cup \{\mathbf{y}_k\}_{k\leq n} \cup \{\boldsymbol{\Theta}\}$ on the measurable space $(\sigma(\mathbf{x}_{0:n}, \mathbf{y}_{1:n}, \boldsymbol{\Theta}), \mathcal{X}^{n+1} \times \mathcal{Y}^n \times \mathsf{S})$.

function $f : \mathcal{X} \to \mathbb{R}$,

$$(f, \xi_{n,\boldsymbol{\theta}}) = \int \int f(\mathbf{x}) \tau_{n,\boldsymbol{\theta}}(d\mathbf{x}|\mathbf{x}') \phi_{n-1,\boldsymbol{\theta}}(d\mathbf{x}') = (f, \tau_{n,\boldsymbol{\theta}} \phi_{n-1,\boldsymbol{\theta}}).$$

We also note that

$$(f, \xi_{n,\boldsymbol{\theta}}) = (\bar{f}_n, \phi_{n-1,\boldsymbol{\theta}}),$$

where $\bar{f}_n(\mathbf{x}') = \int f(\mathbf{x}) \tau_{n,\boldsymbol{\theta}}(d\mathbf{x}|\mathbf{x}')$. The filter measures $\phi_{n,\boldsymbol{\theta}}$ and $\phi_{n-1,\boldsymbol{\theta}}$ are related by the projective product

$$\phi_{n,\boldsymbol{\theta}} = u_n^{\mathbf{y}_n} \star \tau_{n,\boldsymbol{\theta}} \phi_{n-1,\boldsymbol{\theta}} = u_n^{\mathbf{y}_n} \star \xi_{n,\boldsymbol{\theta}},$$

defined as [12]

$$(f, u_n^{\mathbf{y}_n} \star \xi_{n,\boldsymbol{\theta}}) \triangleq \frac{(f u_n^{\mathbf{y}_n}, \xi_{n,\boldsymbol{\theta}})}{(u_n^{\mathbf{y}_n}, \xi_{n,\boldsymbol{\theta}})}.$$

Let

$$\xi_{n,\boldsymbol{\theta}}^J(d\mathbf{x}) = \frac{1}{J} \sum_{j=1}^{J} \delta_{\mathbf{x}_n^{(j)}}(d\mathbf{x}) \quad \text{and} \quad \phi_{n,\boldsymbol{\theta}}^J(d\mathbf{x}) = \frac{1}{J} \sum_{j=1}^{J} \delta_{\tilde{\mathbf{x}}_n^{(j)}}(d\mathbf{x})$$

be the approximations of $\xi_{n,\boldsymbol{\theta}}$ and $\phi_{n,\boldsymbol{\theta}}$ produced by a standard PF [73] (see Table 2.1) with J particles. We have the following theoretical guarantee for the convergence of $\xi_{n,\boldsymbol{\theta}}^J$ and $\phi_{n,\boldsymbol{\theta}}^J$.

Lemma 1. *Let N be a finite time horizon and let $\boldsymbol{Y}_{1:N} = \mathbf{y}_{1:N}$ be an arbitrary but fixed sequence of observations. Assume that, for every $n = 1, ..., N$, $u_n^{\mathbf{y}_n} \in B(\mathcal{X})$, S is compact and*

$$\inf_{\boldsymbol{\theta} \in \mathsf{S}} (u_n^{\mathbf{y}_n}, \xi_{n,\boldsymbol{\theta}}) > 0. \tag{4.7}$$

Then, for every $f \in B(\mathcal{X})$, every $p \geq 1$ and every $n = 0, 1, ..., N$,

$$\sup_{\boldsymbol{\theta} \in \mathsf{S}} \|(f, \xi_{n,\boldsymbol{\theta}}^J) - (f, \xi_{n,\boldsymbol{\theta}})\|_p \leq \frac{c_{1,n} \|f\|_\infty}{\sqrt{J}} \tag{4.8}$$

$$\sup_{\boldsymbol{\theta} \in \mathsf{S}} \|(f, \phi_{n,\boldsymbol{\theta}}^J) - (f, \phi_{n,\boldsymbol{\theta}})\|_p \leq \frac{c_{2,n} \|f\|_\infty}{\sqrt{J}}, \tag{4.9}$$

where $c_{1,n}$ and $c_{2,n}$ are positive and finite constants independent of J and $\boldsymbol{\theta}$.

Proof. This is a straightforward consequence of [44, Lemma 2]. □

We denote the likelihood of the parameter realization $\boldsymbol{\theta}$ given the observations $\mathbf{Y}_{1:N} = \mathbf{y}_{1:N}$ as $\lambda_N(\boldsymbol{\theta})$, where

$$\lambda_N(\boldsymbol{\theta}) \triangleq \prod_{n=1}^{N} (u_n^{\mathbf{y}_n}, \xi_{n,\boldsymbol{\theta}})$$

(it is straightforward to show that $\lambda_N(\boldsymbol{\theta})$ yields the value of the joint pdf of $\mathbf{y}_1, \ldots, \mathbf{y}_N$ conditional on $\boldsymbol{\theta}$). This likelihood can be naturally approximated via a PF as

$$\lambda_N^J(\boldsymbol{\theta}) \triangleq \prod_{n=1}^{N} (u_n^{\mathbf{y}_n}, \xi_{n,\boldsymbol{\theta}}^J)$$

and still guarantee that $\lambda_N^J \to \lambda_N$ in MSE with standard Monte Carlo rates [37]. This is examined below.

4.4.1 Particle approximation of the parameter likelihood

We proceed with a slight variation of the result on the MSE convergence of $\lambda_N^J(\boldsymbol{\theta})$ proved in [37]. This requires to strengthen some assumptions to make them uniform on $\mathsf{S} \subset \mathbb{R}^K$.

Assumption 4. *There exists a constant $a > 0$ such that $\inf_{\mathbf{x}\in\mathcal{X}} u_n^{\mathbf{y}_n}(\mathbf{x}) \geq 1/a > 0$ for every $n = 1, \ldots, N$. Moreover, $u_n^{\mathbf{y}_n} \in B(\mathcal{X})$ for every $n = 1, \ldots, N$ as well.*

Remark 2. *Recall that $u_n^{\mathbf{y}_n}$ is independent of $\boldsymbol{\theta} \in \mathsf{S}$ for every $n = 1, \ldots, N$.*

Remark 3. *Assumption 4 implies that $R_n = \sup_{(\mathbf{x},\mathbf{x}')\in\mathcal{X}^2} \frac{u_n^{\mathbf{y}_n}(\mathbf{x})}{u_n^{\mathbf{y}_n}(\mathbf{x}')} < \infty$ for $n = 1, \ldots, N$.*

Definition 1. *The m-steps-ahead transition kernel $\tau_{n:n+m,\boldsymbol{\theta}}$ is constructed as*

$$\tau_{n:n+m,\boldsymbol{\theta}}(d\mathbf{x}_{n+m}|\mathbf{x}_n) = \int \ldots \int \begin{array}{l} \tau_{n+m}(d\mathbf{x}_{n+m}|\mathbf{x}_{n+m-1}) \\ \tau_{m+n-1}(d\mathbf{x}_{n+m-1}|\mathbf{x}_{n+m-2}) \\ \vdots \\ \tau_{n+1}(d\mathbf{x}_{n+1}|\mathbf{x}_n). \end{array}$$

Assumption 5. *There exists $m \in \mathbb{N}$, and some sequence of numbers $1 \leq \beta_p^{(m)} < \infty$ independent of $\boldsymbol{\theta}$, such that for any $p \geq 0$ and any $(\mathbf{x}, \mathbf{x}') \in \mathcal{X}^2$ we have*

$$\tau_{p:p+m,\boldsymbol{\theta}}(d\mathbf{y}|\mathbf{x}) \leq \beta_p^{(m)} \tau_{p:p+m,\boldsymbol{\theta}}(d\mathbf{y}|\mathbf{x}').$$

Assumptions 4 and 5 lead to the following MSE convergence results for λ_N^J.

Lemma 2. *If Assumptions 4 and 5 hold for some $(m, \{R_n\}_{1\leq n\leq N}, \{\beta_p^{(m)}\}_{p\geq 0})$ then, for every $J > \sum_{s=0}^N R_q^{(m)} \beta_s^{(m)}$, and any $\boldsymbol{\theta} \in \mathsf{S}$,*

$$\mathbb{E}\left[\left(\lambda_N^J(\boldsymbol{\theta}) - \lambda_N(\boldsymbol{\theta})\right)^2\right] \leq \frac{1}{J}\left(4\lambda_N(\boldsymbol{\theta}) \sum_{s=0}^N R_q^{(m)} \beta_s^{(m)}\right),$$

where $R_q^{(m)} = \prod_{s\leq q\leq s+m} R_q$.

Proof. See [37], Corollary 5.2. □

Lemma 3. *If Assumptions 4 and 5 hold for some $(m, \{R_n\}_{1\leq n\leq N}, \{\beta_p^{(m)}\}_p)$, then, for sufficiently large J,*

$$\mathbb{E}\left[\left(\lambda_N^J(\boldsymbol{\theta}) - \lambda_N(\boldsymbol{\theta})\right)^2\right] \leq \frac{c_N}{J},$$

where $c_N < \infty$ is a constant independent of $\boldsymbol{\theta}$ and J.

Proof. Simply recall Assumption 4, which particularly yields

$$\|\lambda_N\|_\infty \leq \prod_{n=1}^N \|u_n^{\mathbf{y}_n}\|_\infty < \infty.$$

Then from Lemma 2, $c_N = 4\left(\prod_{n=1}^N \|u_n^{\mathbf{y}_n}\|_\infty\right) \sum_{s=0}^N R_q^{(m)} \beta_s^{(m)}$, which is independent of $\boldsymbol{\theta}$, as the sequence $\beta_s^{(m)}$ is independent of $\boldsymbol{\theta}$ by virtue of Assumption 5. □

4.4.2 Convergence of NIS with approximate weights

We can put the previous Lemmas together to prove convergence of the NIS scheme with approximate weights.

Assume that we use NIS to approximate the posterior measure of the parameter $\boldsymbol{\theta}$ at time N, namely

$$\pi(\boldsymbol{\theta})d\boldsymbol{\theta} = \mathbb{P}_N\left(\boldsymbol{\Theta} \in d\boldsymbol{\theta} | \mathbf{Y}_{1:N} = \mathbf{y}_{1:N}\right). \tag{4.10}$$

It is straightforward to show that

$$\pi(\boldsymbol{\theta}) \propto h(\boldsymbol{\theta}) = \lambda_N(\boldsymbol{\theta}) m_0(\boldsymbol{\theta}),$$

where $m_0(\boldsymbol{\theta})$ is the density associated to the prior probability distribution of the parameter, μ_0. If a proposal pdf q is used, the weight function becomes

$$g(\boldsymbol{\theta}) = \frac{h(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} = \frac{\lambda_N(\boldsymbol{\theta}) m_0(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}. \tag{4.11}$$

Since the likelihood $\lambda_N(\boldsymbol{\theta})$ cannot be computed in closed form we readily approximate it using a PF. This, in turn, yields the approximate weight function

$$g^J(\boldsymbol{\theta}) = \frac{h^J(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} = \frac{\lambda_N^J(\boldsymbol{\theta}) m_0(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}. \tag{4.12}$$

Let us apply a NIS scheme to approximate the target distribution in (4.10), where the weight function can be approximately evaluated using (4.12). The approximation of π with standard IWs is denoted $\pi^{M,J}$ and the approximation with TIWs is denoted $\bar{\pi}^{M,J}$. The observations $\mathbf{y}_{1:N}$ are arbitrary but fixed. Then we have the following result.

Theorem 2. *Assume that*

(i) $J = J(M) \geq M$ *and* $M_T \leq \sqrt{M}$*;*

(ii) Assumptions 4 and 5 hold for some $(m, \{R_n\}_{1 \leq n \leq N}, \{\beta_p^{(m)}\}_{p \in \mathbb{N}})$*;*

(iii) the ratio $\frac{m_0}{q}$ *is uniformly bounded on* S*, i.e.,*

$$\left\| \frac{m_0}{q} \right\|_\infty = \sup_{\boldsymbol{\theta} \in \mathsf{S}} \left| \frac{m_0(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \right| < \infty$$

and there exists $r_0 > 0$ *such that* $\inf_{\boldsymbol{\theta} \in \mathsf{S}} \frac{m_0(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} \geq \frac{1}{r_0}$.

Then, there exist finite constants $\hat{c}_N$ *and* $\bar{c}_N$ *independent of* M*,* M_T *and* J *such that*

$$\|(f, \pi^{M,J}) - (f, \pi)\|_2 \leq \frac{\hat{c}_N}{\sqrt{M}} \quad \textit{and} \tag{4.13}$$

$$\|(f, \bar{\pi}^{M,J}) - (f, \pi)\|_2 \leq \frac{\bar{c}_N}{\sqrt{M}}, \tag{4.14}$$

for any $f \in B(\mathsf{S})$ *and sufficiently large* J*.*

Proof. See Section 4.5.3. □

4.5 Proofs

4.5.1 Proof of Proposition 1

Let us introduce a new set of (unnormalized) bridge weights of the form

$$\breve{w}^{(i)} = \frac{g(\boldsymbol{\theta}^{(i)})^{\gamma}}{\sum_{j=1}^{M} g(\boldsymbol{\theta}^{(j)})}, \quad i = 1, ..., M, \tag{4.15}$$

and the associated (unnormalized) measure $\breve{\pi}^M(d\boldsymbol{\theta}) = \sum_{i=1}^{M} \breve{w}^{(i)} \delta_{\boldsymbol{\theta}^{(i)}}(d\boldsymbol{\theta})$. Using $\breve{\pi}^M$, the absolute difference $|(f, \pi^M) - (f, \bar{\pi}^M)|$ can be upper bounded by way of the triangular inequality

$$|(f, \pi^M) - (f, \bar{\pi}^M)| \leq |(f, \pi^M) - (f, \breve{\pi}^M)| + |(f, \breve{\pi}^M) - (f, \bar{\pi}^M)|. \tag{4.16}$$

In the sequel, we manipulate the two terms on the right hand side of (4.16) to show that (4.2) holds.

From the definition of the bridge weights in (4.15), we obtain that

$$\begin{aligned}
(f, \pi^M) - (f, \breve{\pi}^M) &= \sum_{i=1}^{M} f(\boldsymbol{\theta}^{(i)}) \frac{g(\boldsymbol{\theta}^{(i)}) - g(\boldsymbol{\theta}^{(i)})^{\gamma}}{\sum_{j=1}^{M} g(\boldsymbol{\theta}^{(j)})} \\
&= \sum_{i=1}^{M} f(\boldsymbol{\theta}^{(i)}) \frac{g(\boldsymbol{\theta}^{(i)})(1 - g(\boldsymbol{\theta}^{(i)})^{\gamma-1})}{\sum_{j=1}^{M} g(\boldsymbol{\theta}^{(j)})} \\
&= (f(1 - g^{\gamma-1}), \pi^M),
\end{aligned} \tag{4.17}$$

where the last equality follows trivially if we recall the standard weight function $w^{(i)} = g(\boldsymbol{\theta}^{(i)}) / \sum_{j=1}^{M} g(\boldsymbol{\theta}^{(j)})$.

As for the second term on the right hand side of (4.16), the definitions of $\bar{w}^{(i)}$ and $\breve{w}^{(i)}$ in (4.1) and (4.15), respectively, yield

$$\begin{aligned}
(f, \breve{\pi}^M) - (f, \bar{\pi}^M) &= \sum_{i=1}^{M} f(\boldsymbol{\theta}^{(i)}) g(\boldsymbol{\theta}^{(i)})^{\gamma} \\
&\times \left(\frac{1}{\sum_{j=1}^{M} g(\boldsymbol{\theta}^{(j)})} - \frac{1}{\sum_{j=1}^{M} g(\boldsymbol{\theta}^{(j)})^{\gamma}} \right).
\end{aligned} \tag{4.18}$$

Some straightforward manipulations show that the difference of fractions above can be rewritten as

$$\begin{aligned}
\frac{1}{\sum_{j=1}^{M} g(\boldsymbol{\theta}^{(j)})} - \frac{1}{\sum_{j=1}^{M} g(\boldsymbol{\theta}^{(j)})^{\gamma}} &= \frac{\sum_{r=1}^{M} g(\boldsymbol{\theta}^{(r)}) \left(g(\boldsymbol{\theta}^{(r)})^{\gamma-1} - 1 \right)}{\sum_{j=1}^{M} g(\boldsymbol{\theta}^{(j)})^{\gamma} \sum_{k=1}^{M} g(\boldsymbol{\theta}^{(k)})} \\
&= \frac{(g^{\gamma-1} - 1, \pi^M)}{\sum_{j=1}^{M} g(\boldsymbol{\theta}^{(j)})^{\gamma}},
\end{aligned} \tag{4.19}$$

where we have used, again, the definition of the standard weights $w^{(i)} = g(\boldsymbol{\theta}^{(i)})/\sum_{j=1}^{M} g(\boldsymbol{\theta}^{(j)})$. Substituting (4.19) into (4.18), and using the definition of TIWs given by (4.1), yields

$$\begin{aligned}(f, \breve{\pi}^M) - (f, \bar{\pi}^M) &= \sum_{i=1}^{M} f(\boldsymbol{\theta}^{(i)})\bar{w}^{(i)}(g^{\gamma-1} - 1, \pi^M) \\ &= (f, \bar{\pi}^M)(g^{\gamma-1} - 1, \pi^M). \end{aligned} \tag{4.20}$$

Finally, substituting (4.20) and (4.17) into (4.16) we arrive at

$$|(f, \pi^M) - (f, \bar{\pi}^M)| \leq |(f(1 - g^{\gamma-1}), \pi^M)| + |(f, \bar{\pi}^M)||(g^{\gamma-1} - 1, \pi^M)|,$$

and the proof concludes by simply noting that $|(f, \bar{\pi}^M)| \leq \|f\|_\infty$ and $|(g^{\gamma-1} - 1, \pi^M)| = |(1 - g^{\gamma-1}, \pi^M)|$. □

4.5.2 Proof of Theorem 1

We consider the approximate integral $(f, \pi^{M,\epsilon})$ first. Since

$$(f, \pi) = \frac{(fg, q)}{(g, q)} \quad \text{and} \quad (f, \pi^{M,\epsilon}) = \frac{(fg^\epsilon, q^M)}{(g^\epsilon, q^M)}, \tag{4.21}$$

where $q^M = \frac{1}{M}\sum_{i=1}^{M} \delta_{\boldsymbol{\theta}^{(i)}}$, it is simple to show that

$$(f, \pi^{M,\epsilon}) - (f, \pi) = \frac{(fg^\epsilon, q^M) - (fg, q)}{(g, q)} + (f, \pi^{M,\epsilon})\frac{(g, q) - (g^\epsilon, q^M)}{(g, q)}. \tag{4.22}$$

However, since $(g, q) = (1, h) = \int I_{\mathsf{S}}(\boldsymbol{\theta})h(\boldsymbol{\theta})d\boldsymbol{\theta}$ and $(f, \pi^{M,\epsilon}) \leq \|f\|_\infty$, equation (4.22) readily yields

$$|(f, \pi^{M,\epsilon}) - (f, \pi)| \leq \frac{1}{(1, h)}\left|(fg^\epsilon, q^M) - (fg, q)\right| + \frac{\|f\|_\infty}{(1, h)}\left|(g, q) - (g^\epsilon, q^M)\right|, \tag{4.23}$$

and, therefore, the problem reduces to computing bounds for errors of the form $|(bg^\epsilon, q^M) - (bg, q)|$, where $b \in B(\mathsf{S})$.

Choose any $b \in B(\mathsf{S})$. A simple triangle inequality yields

$$|(bg^\epsilon, q^M) - (bg, q)| \leq |(bg^\epsilon, q^M) - (bg, q^M)| + |(bg, q^M) - (bg, q)|. \tag{4.24}$$

Since $q^M = \frac{1}{M}\sum_{i=1}^{M} \delta_{\boldsymbol{\theta}^{(i)}}$, for the second term on the right hand side of (4.24) we can write

$$\mathbb{E}\left[|(bg, q^M) - (bg, q)|^p\right] = \mathbb{E}\left[\left|\frac{1}{M}\sum_{i=1}^{M} Z^{(i)}\right|^p\right], \tag{4.25}$$

for arbitrary $p \geq 1$, where the random variables

$$Z^{(i)} = b(\boldsymbol{\theta}^{(i)})g(\boldsymbol{\theta}^{(i)}) - (bg, q), \quad i = 1, ..., M,$$

are i.i.d. with zero mean (recall the $\boldsymbol{\theta}^{(i)}$'s are i.i.d. draws from q). Therefore, it is straightforward to show that

$$\mathbb{E}\left[\left|\frac{1}{M}\sum_{i=1}^{M} Z^{(i)}\right|^p\right] \leq \frac{\tilde{c}^p a^p \|b\|_\infty^p}{M^{\frac{p}{2}}}, \tag{4.26}$$

where $\tilde{c}$ is a constant independent of M and q, and a is the uniform upper bound for the weight function g provided by Assumption 2, also independent of M. Combining (4.26) with (4.25) readily yields

$$\|(bg, q^M) - (bg, q)\|_p \leq \frac{\tilde{c}a\|b\|_\infty}{\sqrt{M}}. \tag{4.27}$$

The inequality (4.27) implies that there exists an a.s. finite random variable $U_{\upsilon,b} > 0$ such that

$$|(bg, q^M) - (bg, q)| \leq \frac{U_{\upsilon,b}}{M^{\frac{1}{2}-\upsilon}}, \tag{4.28}$$

where $0 < \upsilon < \frac{1}{2}$ is an arbitrarily small constant independent of M (see [45, Lemma 4.1]).

If we expand the first term on the right hand side of (4.24) we arrive at

$$\begin{aligned} \left|(bg^\epsilon, q^M) - (bg, q^M)\right| &= \left|\frac{1}{M}\sum_{i=1}^{M} b(\boldsymbol{\theta}^{(i)})\left(g^\epsilon(\boldsymbol{\theta}^{(i)}) - g(\boldsymbol{\theta}^{(i)})\right)\right| \\ &\leq \frac{\|b\|_\infty}{M}\sum_{i=1}^{M}\left|g^\epsilon(\boldsymbol{\theta}^{(i)}) - g(\boldsymbol{\theta}^{(i)})\right|. \end{aligned} \tag{4.29}$$

However, using assumption Assumption 1 in the inequality (4.29) above, we readily obtain

$$\left|(bg^\epsilon, q^M) - (bg, q^M)\right| \leq \|b\|_\infty \epsilon. \tag{4.30}$$

Taking together (4.24), (4.28) and (4.30) we arrive at

$$|(bg^\epsilon, q^M) - (bg, q)| \leq \|b\|_\infty \epsilon + \frac{U_{\upsilon,b}}{M^{\frac{1}{2}-\upsilon}} \tag{4.31}$$

and it is immediate to combine the inequality (4.23) with the bound in (4.31). If we choose $b = f$ in order to control the first term on the right

hand side of (4.23), and $b = 1$ in order to control the second term, we readily find that

$$|(f, \pi^{M,\epsilon}) - (f, \pi)| \leq \frac{W_{f,\upsilon}}{M^{\frac{1}{2}-\upsilon}} + \frac{2\|f\|_\infty}{(1,h)}\epsilon, \tag{4.32}$$

where

$$W_{f,\upsilon} = \frac{U_{\upsilon,f} + U_{\upsilon,1}}{(1,h)} > 0$$

is an a.s. finite random variable independent of M and ϵ. This yields the inequality (4.5) in the statement of Theorem 1, with $C = 2\|f\|_\infty/(1,h) < \infty$ (note that $(1,h) > 0$, see Section 4.1).

The proof for inequality (4.6) is simpler. A triangle inequality yields

$$|(f, \bar{\pi}^{M,\epsilon}) - (f, \pi)| \leq |(f, \bar{\pi}^{M,\epsilon}) - (f, \pi^{M,\epsilon})| + |(f, \pi^{M,\epsilon}) - (f, \pi)| \tag{4.33}$$

and (4.32) yields a bound for the second term on the right hand side of (4.33). For the first term, we note that

$$(f, \bar{\pi}^{M,\epsilon}) = \frac{(f[\varphi^M \circ g^\epsilon], q^M)}{(\varphi^M \circ g^\epsilon, q^M)}, \tag{4.34}$$

where $\circ$ denotes composition, hence $(\varphi^M \circ g^\epsilon)(\boldsymbol{\theta}) = \varphi^M(g^\epsilon(\boldsymbol{\theta}))$. If we combine (4.34) and the expression for $(f, \pi^{M,\epsilon})$ in (4.21) we obtain, by the same argument leading to (4.23), that

$$\begin{aligned} |(f, \bar{\pi}^{M,\epsilon}) - (f, \pi^{M,\epsilon})| &\leq \frac{|(f[\varphi^M \circ g^\epsilon], q^M) - (fg^\epsilon, q^M)|}{(\varphi^M \circ g^\epsilon, q^M)} \\ &\quad + \frac{\|f\|_\infty |(\varphi^M \circ g^\epsilon, q^M) - (g^\epsilon, q^M)|}{(\varphi^M \circ g^\epsilon, q^M)} \\ &\leq a|(f[\varphi^M \circ g^\epsilon], q^M) - (fg^\epsilon, q^M)| \\ &\quad + a\|f\|_\infty |(\varphi^M \circ g^\epsilon, q^M) - (g^\epsilon, q^M)|, \end{aligned} \tag{4.35}$$

where the second inequality follows from the definition of the clipping transformation φ^M and the bound $g^\epsilon \geq a^{-1}$ in Assumption 3.

In order to use (4.35), we look into errors of the form $|(b[\varphi^M \circ g^\epsilon], q^M) - (bg^\epsilon, q^M)|$ for arbitrary $b \in B(\mathsf{S})$. This turns out relatively straightforward since, from the construction of φ^M,

$$\begin{aligned} |(b[\varphi^M \circ g^\epsilon], q^M) - (bg^\epsilon, q^M)| &= \left| \frac{1}{M} \sum_{r=1}^{M_T} b(\boldsymbol{\theta}^{(i_r)}) \left[g^\epsilon(\boldsymbol{\theta}^{(i_{M_T})}) - g^\epsilon(\boldsymbol{\theta}^{(i_r)}) \right] \right| \\ &\leq 2a\|b\|_\infty \frac{M_T}{M}, \end{aligned} \tag{4.36}$$

where the inequality follows from the bound $g^\epsilon \leq a$ in Assumption 3. We can plug (4.36) into (4.35) twice, first choosing $b = f$ and then $b = 1$, in order to control the two terms in the triangle inequality. As a result, we arrive at the *deterministic* bound

$$|(f, \bar{\pi}^{M,\epsilon}) - (f, \pi^{M,\epsilon})| \leq \frac{2a^2\|f\|_\infty M_T}{M} \leq \frac{2a^2\|f\|_\infty}{\sqrt{M}}, \tag{4.37}$$

where the second inequality follows from the assumption $M_T \leq \sqrt{M}$ in the statement of Theorem 1.

Plugging (4.37) and (4.32) into (4.33) yields

$$|(f, \bar{\pi}^{M,\epsilon}) - (f, \pi)| \leq \frac{W_{f,\upsilon} + 2a^2\|f\|_\infty}{M^{\frac{1}{2}-\upsilon}} + \frac{2\|f\|_\infty}{(1,h)}\epsilon, \tag{4.38}$$

which reduces to the inequality (4.6) in the statement of Theorem 1, with $\bar{W}_{f,\upsilon} = W_{f,\upsilon} + 2a^2\|f\|_\infty > 0$ an a.s. finite random variable and $C = 2\|f\|_\infty/(1,h) < \infty$ a constant, both independent of M, M_T and ϵ. □

4.5.3 Proof of Theorem 2

We look into $(f, \pi^{M,J})$ first. Since

$$(f, \pi) = \frac{(fg, q)}{(g, q)} \text{ and } (f, \pi^{M,J}) = \frac{(fg^J, q^M)}{(g^J, q^M)}, \tag{4.39}$$

where $q^M = \frac{1}{M}\sum_{i=1}^M \delta_{\boldsymbol{\theta}^{(i)}}$, it is simple to show that

$$(f, \pi^{M,J}) - (f, \pi) = \frac{(fg^J, q^M) - (fg, q)}{(g, q)} + (f, \pi^{J,M})\frac{(g, q) - (g^J, q^M)}{(g, q)}. \tag{4.40}$$

However, since $(g, q) = (1, h) = \int I_{\mathsf{S}}(\boldsymbol{\theta})h(\boldsymbol{\theta})d\boldsymbol{\theta}$ and $(f, \pi^{J,M}) \leq \|f\|_\infty$, equation (4.40) readily yields

$$\begin{aligned} |(f, \pi^{M,J}) - (f, \pi)| \quad \leq \quad & \frac{1}{(1,h)}\left|(fg^J, q^M) - (fg, q)\right| \\ & + \frac{\|f\|_\infty}{(1,h)}\left|(g, q) - (g^J, q^M)\right|, \end{aligned} \tag{4.41}$$

and, therefore, the problem reduces to computing L_2 bounds for errors of the form $|(bg^J, q^M) - (bg, q)|$, where $b \in B(\mathsf{S})$.

Choose any $b \in B(\mathsf{S})$. A simple triangle inequality yields

$$|(bg^J, q^M) - (bg, q)| \leq |(bg^J, q^M) - (bg, q^M)| + |(bg, q^M) - (bg, q)|. \quad (4.42)$$

Since $q^M = \frac{1}{M}\sum_{i=1}^{M} \delta_{\boldsymbol{\theta}^{(i)}}$, for the second term on the right hand side of (4.42) we can write

$$\mathbb{E}\left[|(bg, q^M) - (bg, q)|^2\right] = \mathbb{E}\left[\left|\frac{1}{M}\sum_{i=1}^{M} Z^{(i)}\right|^2\right], \quad (4.43)$$

where the random variables

$$Z^{(i)} = b(\boldsymbol{\theta}^{(i)})g(\boldsymbol{\theta}^{(i)}) - (bg, q), \quad i = 1, ..., M,$$

are i.i.d. with zero mean (since the $\boldsymbol{\theta}^{(i)}$'s are i.i.d. draws from q). Therefore, it is straightforward to show that

$$\mathbb{E}\left[\left|\frac{1}{M}\sum_{i=1}^{M} Z^{(i)}\right|^2\right] \leq \frac{\tilde{c}^2 \|g\|_\infty^2 \|b\|_\infty^2}{M}, \quad (4.44)$$

where $\tilde{c}$ is a constant independent of M and $\|g\|_\infty = \sup_{\boldsymbol{\theta}\in\mathsf{S}} |g(\boldsymbol{\theta})|$.

A finite upper bound for $\|g\|_\infty$ can be easily found. Indeed,

$$\begin{aligned} \|g\|_\infty &= \sup_{\boldsymbol{\theta}\in\mathsf{S}} \left|\frac{\lambda_N(\boldsymbol{\theta}) m_0(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}\right| \\ &\leq \left\|\frac{m_0}{q}\right\|_\infty \sup_{\boldsymbol{\theta}\in\mathsf{S}} |\lambda_N(\boldsymbol{\theta})| \\ &\leq \left\|\frac{m_0}{q}\right\|_\infty \prod_{n=1}^{N} \|u_n^{\mathbf{y}_n}\|_\infty, \end{aligned} \quad (4.45)$$

where the first inequality follows from assumption (iii) in the statement of Theorem 2, and the second inequality is a consequence of Assumption 4. We readily combine equations (4.43), (4.44) and (4.45) to arrive at

$$\|(bg, q^M) - (bg, q)\|_2 \leq \frac{\tilde{c}\|b\|_\infty \left\|\frac{m_0}{q}\right\|_\infty \prod_{n=1}^{N} \|u_n^{\mathbf{y}_n}\|_\infty}{\sqrt{M}}, \quad (4.46)$$

where the numerator is constant w.r.t. the number of samples M.

In order to control the first term on the right hand side of (4.42), let us expand it to obtain

$$\begin{aligned} \left|(bg^J, q^M) - (bg, q^M)\right| &= \left|\frac{1}{M}\sum_{i=1}^{M} b(\boldsymbol{\theta}^{(i)})\left(g^J(\boldsymbol{\theta}^{(i)}) - g(\boldsymbol{\theta}^{(i)})\right)\right| \qquad (4.47) \\ &\leq \left\|\frac{m_0}{q}\right\|_\infty \left|\frac{1}{M}\sum_{i=1}^{M} b(\boldsymbol{\theta}^{(i)})(\lambda_N^J(\boldsymbol{\theta}^{(i)}) - \lambda_N(\boldsymbol{\theta}^{(i)}))\right|, \end{aligned}$$

where the inequality is obtained by recalling that $g = \frac{m_0\lambda_N}{q}$, $g^J = \frac{m_0\lambda_N^J}{q}$ (see equations (4.11) and (4.12)) and applying assumption (iii) in the statement of Theorem 2. Equation (4.47) readily yields an upper bound for the MSE of $(bg^J, q^M) - (bg, q^M)$ of the form

$$\begin{aligned} & \mathbb{E}\left[\left|(bg^J, q^M) - (bg, q^M)\right|^2\right] \\ \leq\ & \left\|\frac{m_0}{q}\right\|_\infty^2 \mathbb{E}\left[\left|\frac{1}{M}\sum_{i=1}^{M} b(\boldsymbol{\theta}^{(i)})(\lambda_N^J(\boldsymbol{\theta}^{(i)}) - \lambda_N(\boldsymbol{\theta}^{(i)}))\right|^2\right] \\ \leq\ & \frac{\|b\|_\infty^2 \left\|\frac{m_0}{q}\right\|_\infty^2}{M}\sum_{i=1}^{M}\mathbb{E}\left[\left|\lambda_N^J(\boldsymbol{\theta}^{(i)}) - \lambda_N(\boldsymbol{\theta}^{(i)})\right|^2\right], \qquad (4.48) \end{aligned}$$

where the second line results from the application of Jensen's inequality, which yields

$$\left|\frac{1}{M}\sum_{i=1}^{M} b(\boldsymbol{\theta}^{(i)})(\lambda_N^J(\boldsymbol{\theta}^{(i)}) - \lambda_N(\boldsymbol{\theta}^{(i)}))\right|^2 \leq \frac{\|b\|_\infty^2}{M}\sum_{i=1}^{M}\left|\lambda_N^J(\boldsymbol{\theta}^{(i)}) - \lambda_N(\boldsymbol{\theta}^{(i)})\right|^2.$$

Applying[2] Lemma 3 on the right hand side of equation (4.48) we arrive at the inequality

$$\mathbb{E}\left[\left|(bg^J, q^M) - (bg, q^M)\right|^2\right] \leq \frac{c_N\|b\|_\infty^2\left\|\frac{m_0}{q}\right\|_\infty^2}{J}, \qquad (4.49)$$

where $c_N = 4(\prod_{n=1}^{N}\|u_n^{\mathbf{y}_n}\|_\infty)\sum_{s=0}^{N}\beta_s^{(m)}(\prod_{s\leq q\leq s+m}R_q)$ is a constant independent of J and $\boldsymbol{\theta}$ (namely $\sup_{1\leq i\leq M}\mathbb{E}[|\lambda_N^J(\boldsymbol{\theta}^{(i)}) - \lambda_N(\boldsymbol{\theta}^{(i)})|^2] \leq \frac{c_N}{J}$),

[2]The assumptions in Lemmas 2 and 3 are a subset of the assumptions of Theorem 2.

that holds for sufficiently large J. From (4.49) and the assumption $J \geq M$ we readily obtain

$$\left\|(bg^J, q^M) - (bg, q^M)\right\|_2 \leq \frac{\sqrt{c_N}\|b\|_\infty \left\|\frac{m_0}{q}\right\|_\infty}{\sqrt{M}}. \tag{4.50}$$

Now, equations (4.42), (4.46) and (4.50) together yield, via Minkowski's inequality,

$$\left\|(bg^J, q^M) - (bg, q)\right\|_2 \leq \frac{\tilde{c}_N\|b\|_\infty}{\sqrt{M}}, \tag{4.51}$$

where $\tilde{c}_N = (\sqrt{c_N} + \tilde{c}\prod_{n=1}^N \|u_n^{\mathbf{y}_n}\|_\infty)\|\frac{m_0}{q}\|_\infty < \infty$ is a constant independent of M and J. Combining (4.51) with the triangle inequality (4.41) (take $b = f$ for the first term and $b = 1$ for the second term) we obtain

$$\|(f, \pi^{M,J}) - (f, \pi)\|_2 \leq \frac{\hat{c}_N}{\sqrt{M}}, \tag{4.52}$$

where $\hat{c}_N = \frac{2\tilde{c}_N\|f\|_\infty}{(1,h)} < \infty$ is, again, constant w.r.t. M (and J).

The proof for the second inequality in the statement of Theorem 2 is simpler. A triangle inequality yields

$$|(f, \bar{\pi}^{M,J}) - (f, \pi)| \leq |(f, \bar{\pi}^{M,J}) - (f, \pi^{M,J})| + |(f, \pi^{M,J}) - (f, \pi)| \tag{4.53}$$

and we have already found an adequate bound for the L_2 norm of the second term on the right hand side of (4.53). For the first term on the right hand side, we note that

$$(f, \bar{\pi}^{M,J}) = \frac{(f[\varphi^M \circ g^J], q^M)}{(\varphi^M \circ g^J, q^M)}, \tag{4.54}$$

where $\circ$ denotes composition, hence $(\varphi^M \circ g^J)(\boldsymbol{\theta}) = \varphi^M(g^J(\boldsymbol{\theta}))$. Taking together (4.54) and the expression for $(f, \pi^{M,J})$ in (4.39) yields, by the same argument leading to (4.41),

$$\begin{aligned} |(f, \bar{\pi}^{M,J}) - (f, \pi^{M,J})| \quad &\leq \quad \frac{|(f[\varphi^M \circ g^J], q^M) - (fg^J, q^M)|}{(\varphi^M \circ g^J, q^M)} \\ &\quad + \frac{\|f\|_\infty |(\varphi^M \circ g^J, q^M) - (g^J, q^M)|}{(\varphi^M \circ g^J, q^M)} \\ &\leq \quad \left[|(f[\varphi^M \circ g^J], q^M) - (fg^J, q^M)|\right. \\ &\quad \left. + \|f\|_\infty |(\varphi^M \circ g^J, q^M) - (g^J, q^M)|\right] ar_0, \end{aligned} \tag{4.55}$$

where the second inequality follows from the definition of φ^M, Assumption 4 and assumption (iii) in the statement of Theorem 2.

In order to use (4.55), we look into errors of the form $|(b[\varphi^M \circ g^J], q^M) - (bg^J, q^M)|$ for arbitrary $b \in B(\mathsf{S})$. This turns out relatively straightforward since, from the definition of φ^M (recall equation (4.4)),

$$\begin{aligned} |(b[\varphi^M \circ g^J], q^M) - (bg^J, q^M)| &= \\ \left| \frac{1}{M} \sum_{r=1}^{M_T} b(\boldsymbol{\theta}^{(i_r)}) \left[\frac{m_0(\boldsymbol{\theta}^{(i_{M_T})})}{q(\boldsymbol{\theta}^{(i_{M_T})})} \lambda_N^J(\boldsymbol{\theta}^{(i_{M_T})}) - \frac{m_0(\boldsymbol{\theta}^{(i_r)})}{q(\boldsymbol{\theta}^{(i_r)})} \lambda_N^J(\boldsymbol{\theta}^{(i_r)}) \right] \right| &\leq \\ 2 \left(\prod_{n=1}^{N} \|u_n^{\mathbf{y}_n}\|_\infty \right) \|b\|_\infty \left\| \frac{m_0}{q} \right\|_\infty \frac{M_T}{M}, \qquad (4.56) \end{aligned}$$

where the inequality follows Assumption 4 and assumption (iii). We can plug (4.56) into (4.55) twice, first choosing $b = f$ and then $b = 1$, in order to control the two terms in the triangle inequality. As a result, we arrive at the *deterministic* bound

$$|(f, \bar{\pi}^{M,J}) - (f, \pi^{M,J})| \leq \frac{\breve{c}_N \|f\|_\infty M_T}{M} \leq \frac{\breve{c}_N \|f\|_\infty}{\sqrt{M}}, \qquad (4.57)$$

where $\breve{c}_N = 4ar_0(\prod_{n=1}^N \|u_n^{\mathbf{y}_n}\|_\infty)\|\frac{m_0}{q}\|_\infty < \infty$ and the second inequality follows from the assumption $M_T \leq \sqrt{M}$ in the statement of the Theorem 2.

Finally, taking together (4.53), (4.52) and (4.57) yields, via Minkowski's inequality,

$$|(f, \bar{\pi}^{M,J}) - (f, \pi^{M,J})| \leq \frac{\bar{c}_N}{\sqrt{M}}, , \qquad (4.58)$$

where $\bar{c}_N = \breve{c}_N \|f\|_\infty + \hat{c}_N < \infty$ is constant w.r.t. M, M_T and J. □

4.6 Summary

In this chapter we have investigated the distortion introduced by the nonlinear transformations of the IWs performed in the NIS scheme proposed in this thesis. We have obtained convergence rates for the tempering and clipping transformations, both with exact and approximate weights. In particular, we have analyzed the distortion introduced by a PF approximation in the computation of the IWs and found upper bounds for the resulting L_2 error in the approximation of integrals of bounded functions w.r.t. the target distribution.

Chapter 5

Numerical examples

In this chapter we present some simulation results that illustrate the performance of the proposed NPMC algorithms for static models of the type described in Section 2.2.1. In particular, in Section 5.1 we address a simple problem where the target distribution is the posterior of a set of parameters in a Gaussian mixture model (GMM). We use this toy example to illustrate the degeneracy problem and to provide a comparison between the basic NPMC algorithm presented in Section 3.3 and some other relevant techniques. In Section 5.2 we present simulation results regarding the adaptive NMPMC algorithm presented in Section 3.4. We show, by computer simulations, how the nonlinear transformation of the IWs dramatically increases the efficiency of the MPMC algorithm. Additionally, the adaptation of the number of mixture components provides a valuable knowledge of the target distribution.

5.1 Toy example: a Gaussian mixture model

In this section we provide numerical results that illustrate the degeneracy problem and the performance of the proposed NPMC scheme applied to the simple GMM example of [32], where the conditional pdf of the observations has the form

$$p(y|\boldsymbol{\theta}) = \rho\mathcal{N}(y;\theta_1,\sigma^2) + (1-\rho)\mathcal{N}(y;\theta_2,\sigma^2) \tag{5.1}$$

and the random vector of interest, $\boldsymbol{\theta} = [\theta_1,\ \theta_2]^\top$, contains the means of the mixture components. The true values of the unknowns are set to $\boldsymbol{\theta}_* = [0, 2]^\top$. The mixture coefficient and the variance of the components are assumed to be known and set to $\rho = 0.2$ and $\sigma^2 = 1$, respectively.

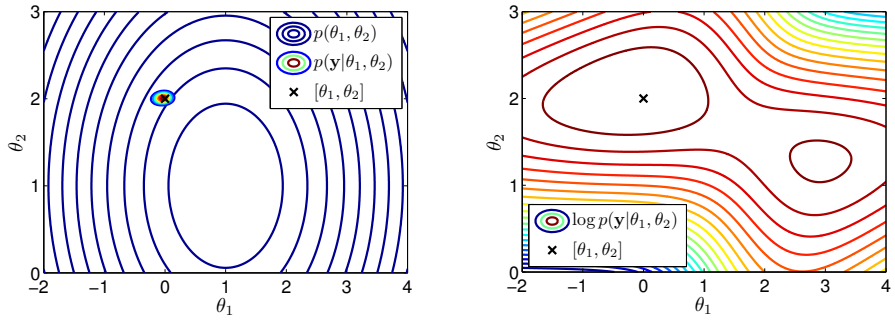

Figure 5.1: *Left*: Contour plot of the prior pdf $p(\boldsymbol{\theta})$ and the likelihood function $p(\mathbf{y}|\boldsymbol{\theta})$ in the GMM example. *Right*: Contour plot of the log likelihood $\log p(\mathbf{y}|\boldsymbol{\theta})$, which reveals the likelihood bimodality.

We assume a prior pdf $p(\boldsymbol{\theta}) = p(\theta_1)p(\theta_2)$ composed of independent components for each unknown, given by $p(\theta_k) = \mathcal{N}(\theta_k; 1, 10)$, $k = 1, 2$. A collection $\mathbf{y}$ of $N = 1000$ i.i.d. scalar observations are drawn from the mixture model in equation (5.1), and we aim at approximating the posterior pdf $p(\boldsymbol{\theta}|\mathbf{y})$.

Figure 5.1 (*left*) depicts the bidimensional prior pdf $p(\boldsymbol{\theta})$ and the likelihood function $p(\mathbf{y}|\boldsymbol{\theta})$ of the parameter vector $\boldsymbol{\theta}$ given a fixed observation vector $\mathbf{y}$. The likelihood function concentrates in a small region of the parameter space represented by the prior pdf, and it is centered on the true value of the parameter $\boldsymbol{\theta}$. However, the logarithm of the likelihood function represented in Figure 5.1 (*right*) reveals that it actually presents a second mode around $\boldsymbol{\theta} = [3, 1.3]^\top$, which has a much lower amplitude.

5.1.1 Performance of the MH algorithm

In this section we show the performance of the MH algorithm, described in Section 2.5.1, when applied to this simple inference problem. MCMC algorithms have been traditionally preferred for Bayesian inference is static models. We illustrate the problematic associated to the MH method in particular, and MCMC algorithms in general.

We have applied the standard MH algorithm of Table 2.6 with a Gaussian random walk proposal with a standard deviation σ and $I = 2000$ iterations. To show the dependance of the performance of this method on the selection of the parameter σ, we have performed simulations with values of σ between 10^{-2} and 10^{1}. We have discarded the first 400 samples as a burn-in period

and performed thinning by a factor of 8, which yields a final sample of size $M = 200$. The results have been averaged over $P = 10^4$ independent simulation runs with different observations, for each value of σ.

In Figure 5.2 (*left*) the average NESS and the acceptance rate are represented versus the value of σ. The NESS has been computed as in equation (2.25) from the final sample, after removing the burn-in period and thinning the output. The acceptance rate is the ratio of the total number of samples accepted in each simulation run to the total length of the chain, I. As expected, the acceptance rate decreases monotonically with the variance of the random walk. When big changes are proposed w.r.t. the current value of the chain, they are very likely to be discarded. However, the NESS takes low values both for high and low σ values, and has its maximum around $\sigma = 0.1$, in this particular case. When σ is very low, many samples are accepted but the correlations among them are very high, yielding a low number of "effective" or independent samples. Thus, in view of this result, it seems reasonable to set the variance of the random walk to 0.1, since it optimizes the efficiency of the sampling procedure in terms of sample independence.

In Figure 5.2 (*right*) the average MSE for parameters θ_1 and θ_2 are represented versus σ, together with the corresponding MMSE values, which have been approximated numerically from the true posterior pdf. The MSE of each parameter θ_k has been computed based on the M-size final output (after removing the burn-in period and thinning), as

$$MSE_k = \frac{1}{M}\sum_{i=1}^{M}(\theta_k^{(i)} - \theta_{k*})^2, \quad k \in \{1, ..., K\}. \tag{5.2}$$

It can be observed that σ values around 0.1 yield high MSE values on average, and that a minimum for the MSE is obtained for $\sigma = 2$, approximately, which in turn yields a very low NESS and acceptance rate. Thus, for σ values around 2, the probability of accepting new samples is very low, but those yield a low MSE on average.

In Figure 5.3 we represent the Markov chains obtained in two simulation runs with $\sigma = 0.1$ (*left*) and $\sigma = 2$ (*right*), respectively. Average simulations have been selected, that attained a final MSE close to the global mean MSE for that σ value. In the first case the chain displays relatively good mixing properties and attains a reasonable NESS. However, given the low exploration capabilities of a transition kernel with low variance, the chain gets stuck in the second mode of the target distribution, which has orders of magnitude less likelihood. In the second case the number of accepted

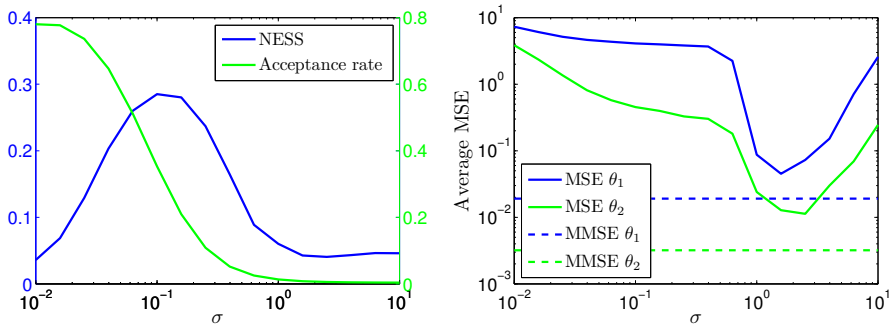

Figure 5.2: Average NESS, acceptance rate (*left*) and MSE (*right*) obtained by the MH algorithm in the GMM example, represented as a function of the random walk standard deviation σ.

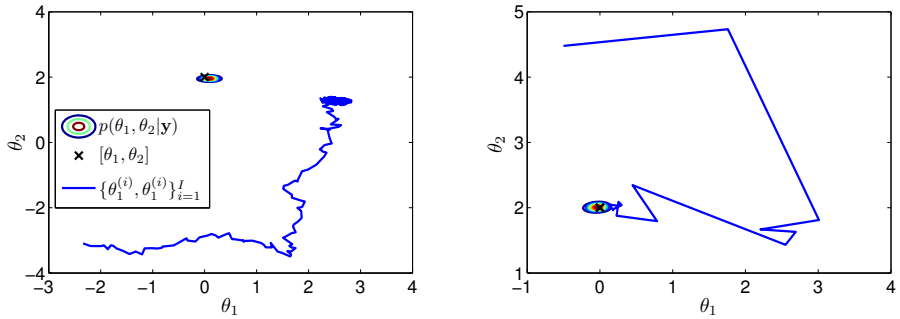

Figure 5.3: Markov chains generated via a MH algorithm in the GMM example. In the *left* plot, the standard deviation of the random walk σ has been set 0.1, to maximize the final average NESS. In the *right* plot, σ has been set to 2, to minimize the final average MSE.

samples is very low and the mixing of the chain is very poor. With this variance selection a much longer chain would be required to obtain good estimates. This sensitivity of the MH algorithm to the selection of the random walk variance is a well known problem that limits its applicability in many practical applications.

5.1.2 Degeneracy of the importance weights

As an alternative to MCMC algorithms, we focus on IS-based methods. In particular, in this section we study the problem of weight degeneracy, described in Section 2.3.2. In order to illustrate the effects of weight

degeneracy, we focus on the plain IS procedure in this simple and low dimensional example.

We consider a set of M samples $\Theta^M = \{\boldsymbol{\theta}^{(i)}\}_{i=1}^M$ drawn form the prior pdf $p(\boldsymbol{\theta})$. Thus, the normalized IWs are computed from the likelihood function as in equation (2.14) and they are of the form

$$w^{(i)} \propto \prod_{n=1}^{N} \rho\mathcal{N}\left(y_n; \theta_1^{(i)}, \sigma^2\right) + (1-\rho)\mathcal{N}\left(y_n; \theta_2^{(i)}, \sigma^2\right).$$

Figure 5.4 illustrates the effect of degeneracy in this simple IS setup. We consider that a set of $N = 10^3$ i.i.d. observations is available from the GMM. A set of $M = 200$ samples have been drawn form the prior pdf and the associated IWs have been computed from the likelihood. The subset of 42 samples closest to the true value of $\boldsymbol{\theta}$ is depicted in Figure 5.4 together with the associated IWs. The likelihood function evaluated in the same region is depicted with contour lines, which show that it is concentrated in a small region of the state space. It can be observed that only a small part of the sample set is close to the region of maximum likelihood. As a result, one sample has a weight close to 1 and the rest of them have negligible IWs.

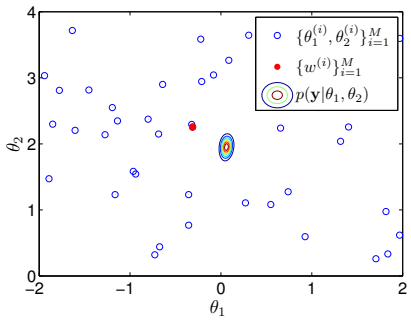

Figure 5.4: Subset of 42 best samples $\boldsymbol{\theta}^{(i)}$ out of $M = 200$ drawn from the prior $p(\boldsymbol{\theta})$ (blue empty circles) and the associated IWs (red filled circles with size proportional to the weight $w^{(i)}$). The likelihood function $p(\mathbf{y}|\boldsymbol{\theta})$ is depicted with contour lines. Due to the narrow likelihood, one sample has weight close to 1 and the rest of them become negligible.

For this model, we have investigated the behavior of the maximum normalized IW, $\max_i w^{(i)}$, and the ESS, M^{eff}, when the number of observations N increases. Let both the number of observations N and the number of samples M vary from 1 to 10^3. For each pair of values of N and

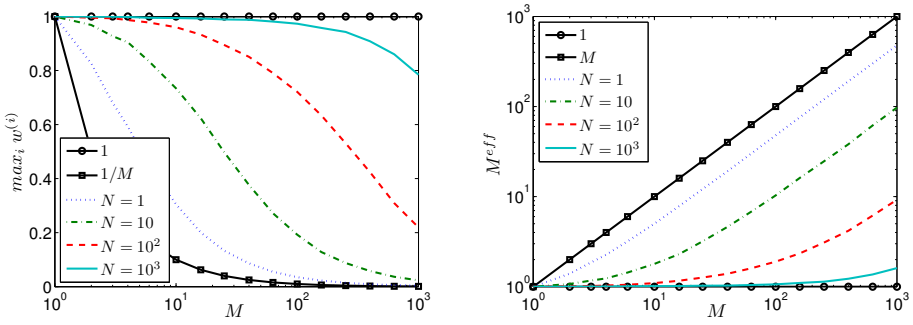

Figure 5.5: Evolution of the average maximum IW, $\max_i w^{(i)}$, (*left*) and the ESS, M^{eff}, (*right*) vs the number of observations, N, and the number of samples, M. The curves corresponding to maximum degeneracy ($\max_i w^{(i)} = 1$ and $M^{eff} = 1$) are plotted with circles. The curves corresponding to the optimum case with uniform weights ($\max_i w^{(i)} = 1/M$ and $M^{eff} = M$) are depicted with squares. All curves are averaged over $P = 10^3$ independent simulation runs.

M, we have performed $P = 10^3$ independent simulation runs of the standard IS procedure, generating a different vector of observations $\mathbf{y}$ and a different set of samples Θ^M in each run.

In Figure 5.5 (*left*) the maximum IW averaged over P simulation runs is represented as a function of the number of samples M and the number of observations N. The curves representing the extreme cases $\max_i w^{(i)} = 1$ (degeneracy) and $\max_i w^{(i)} = 1/M$ (uniform weights) are also plotted on the graph. It can be observed that, for a fixed M, as the number of observations N increases, the maximum IW approaches the extreme degeneracy case $\max_i w^{(i)} \to 1$. This indicates that an increase in the number of observations causes an increase in the variation of the IWs, leading to degeneracy.

Equivalently, in Figure 5.5 (*right*) the average ESS is represented versus M, for several values of N. The cases $M^{eff} = 1$ and $M^{eff} = M$ are plotted for reference. It can be observed that, as N increases, the ESS is smaller for the same value of M. For example, with $N = 10^3$ observations and $M = 10^3$ samples, the average ESS is only 1.5.

These results clearly suggest that in low dimensional problems where the proposal is very broad w.r.t. the target pdf, severe degeneracy can take place. In order to obtain a sufficient ESS by standard IS the generation of an extremely high number of samples is required, which leads to very inefficient schemes with a high computational load.

5.1.3 Illustration of the NPMC algorithm

In this section we illustrate the performance of the NPMC algorithm applied to this simple problem, and we compare it to the DPMC algorithm of [32, 52]. We have performed $P = 10^4$ independent simulation runs of each algorithm, with $L = 10$ iterations and $M = 200$ samples per iteration. The parameters of the DPMC algorithm have been selected as suggested in [32] ($D = 5$ scales, $\boldsymbol{\sigma}^2 = [\sigma_1^2, \ldots, \sigma_D^2]^\top = [5, 2, 0.1, 0.05, 0.01]^\top$), and a minimum of 1% of samples per scale has been kept as a baseline. We have implemented the NPMC method with a clipping transformation with $M_T = 20$.

Figure 5.6 depicts the unweighted sample attained at iterations $\ell = 1$ (*left* column), $\ell = 2$ (*central* column) and $\ell = 4$ (*right* column) of the DPMC algorithm (*upper* row), the PMC method with independent Gaussian proposals and standard IWs (*central* row) and the NPMC with TIWs (*lower* row). On each plot the true posterior pdf is represented, together with the true value of $\boldsymbol{\theta}$ and the set of unweighted samples $\{\tilde{\boldsymbol{\theta}}_\ell^{(i)}\}_{i=1}^M$. It can be seen that the standard IWs at the first iteration are highly degenerate resulting in only two effective samples. The DPMC algorithm is robust to this low ESS and converges to the true posterior in a few iterations, yielding, however, a low final NESS. On the other hand, the PMC method with independent proposals and standard IWs is very sensitive to the low ESS and displays a poor performance. In case the ESS were no higher than 1, this algorithm would yield a numerical error due to the impossibility of constructing the covariance matrix for the next proposal pdf. For this reason, this construction of the proposal pdf has been usually avoided in practical problems. Finally, the NPMC (with independent proposals and TIWs) is robust to the degeneracy problem and smoothly converges to the target pdf reaching a final NESS close to 1.

The DPMC method uses a set of different scales to account for the difficulty of selecting the scale parameter in a random walk proposal (similar to the MCMC problematic). This allows for a global exploration of the parameter space with high σ values, and a local exploration with low σ values. However, this results in the generation of a large number of non representative samples, leading to a low ESS. On the contrary, the proposed NPMC algorithm benefits from the simple independent proposal construction, while effectively addressing the degeneracy problem.

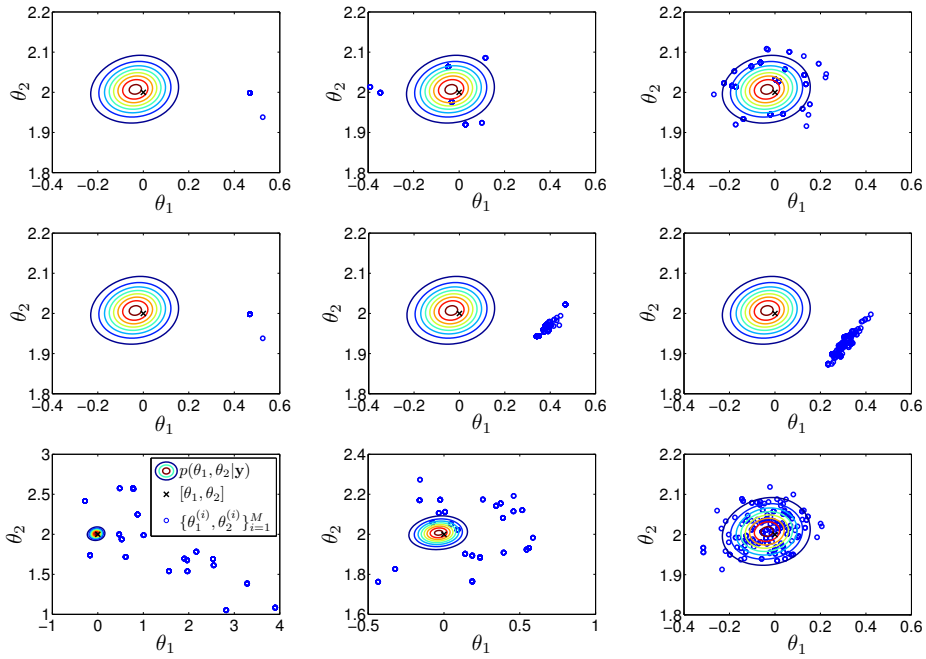

Figure 5.6: True posterior pdf $p(\boldsymbol{\theta}|\mathbf{y})$ and the sample approximations $\{\tilde{\boldsymbol{\theta}}_\ell^{(i)}\}_{i=1}^M$ attained at iterations $\ell = 1, 2, 4$ of the DPMC algorithm (*upper* row), the PMC method with independent proposals and standard IWs (*central* row) and the NPMC with TIWs (*lower* row).

5.1.4 Comparison of PMC and NPMC algorithms

In this section we compare, by way of computer simulations, the performance of the DPMC scheme proposed in [32, 52] and reproduced in Table 2.9, the DPMC with a clipping transformation, and the NPMC scheme of Section 3.3 with tempering and clipping transformations (and Gaussian proposals). We have performed $P = 10^4$ independent simulation runs of each algorithm, with $L = 10$ iterations and $M = 200$ samples per iteration.

The parameters of the DPMC algorithm have again been selected as suggested in [32]. The DPMC scheme with TIWs has been simulated simply substituting the standard IWs $w_\ell^{(i)}$ in the resampling step by TIWs $\bar{w}_\ell^{(i)}$ computed via a clipping transformation (with $M_T = 20$).

In the NPMC algorithm with tempering, the sequence γ_ℓ has been obtained from the sigmoid function of the iteration index as $\gamma_\ell = \frac{1}{1+e^{-(\ell-5)}}$, $\ell = 1, \ldots, L$. With this choice of nonlinearity, the transformation of the

weights is practically eliminated after 10 iterations.

The NPMC algorithm with clipping has been simulated in its modified version, i.e., with the nonlinear transformation removed when the ESS M^{eff} reaches a value of $M_{min}^{eff} = 100$. In this problem this occurs on average between the third and fourth iterations. On the contrary, in the DPMC scheme with clipping, the ESS never reaches the threshold value and the nonlinear transformation thus cannot be removed. The clipping parameter has been set to $M_T = 20$ in both algorithms.

In Figure 5.7 the evolution of the average NESS M_ℓ^{neff} along the iterations is depicted for the DPMC, while $\bar{M}_\ell^{neff}$ is shown for the rest of schemes. It can be observed that the original DPMC scheme presents a low NESS, converging to a value of 0.13. The DPMC with clipping outperforms the original scheme providing an average final NESS of 0.35. The two NPMC schemes, with tempering and clipping, provide a smooth convergence of the NESS to a value of 0.94.

The degeneracy problem is most critical during the first iterations of the PMC algorithm. The DPMC scheme has an initial NESS value close to zero, opposite to the rest of schemes, where $\bar{M}_1^{neff}$ is around 0.1 (it is equal to M_T/M for the clipping schemes and depends on the parameter γ_1 for the tempering scheme). It can be observed from Figure 5.7 that in the NPMC schemes the average NESS remains constant after convergence, when the nonlinear transformation has been removed.

If we interpret the random vector $\boldsymbol{\theta}_\ell$ with distribution $\bar{\pi}_\ell^M(d\boldsymbol{\theta})$ given by equation (3.7) (obtained after the ℓ-th iteration of the NPMC algorithm) as an estimator of $\boldsymbol{\theta}$, then the MSE for the estimator of the k-th parameter θ_k is naturally given by

$$MSE_{\ell,k} = \sum_{i=1}^{M} \bar{w}_\ell^{(i)} (\theta_{\ell,k}^{(i)} - \theta_k)^2 = (\mu_{\ell,k} - \theta_k)^2 + \sigma_{\ell,k}^2, \tag{5.3}$$

where $\mu_{\ell,k}$ is the k-th component of the mean vector $\boldsymbol{\mu}_\ell$ and the variance term $\sigma_{\ell,k}^2$ is the (k,k) component of matrix $\boldsymbol{\Sigma}_\ell$, computed as in equations (3.2) and (3.3), respectively.

In Figure 5.8 the evolution of the MSE for θ_1 ($MSE_{\ell,1}$, *left*) and θ_2 ($MSE_{\ell,2}$, *right*) averaged over P independent simulations is represented for the four algorithms. For the original DPMC algorithm, the MSE has been computed using standard IWs. The MMSE of each parameter are also shown for reference. It can be observed that the DPMC method does not reach the MMSE with the given number of samples $M = 200$. On the other side,

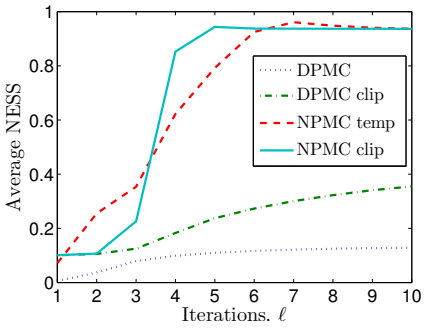

Figure 5.7: Evolution along the iterations of the average NESS for the DPMC, DPMC with clipping, NPMC with tempering and NPMC with clipping for the GMM example.

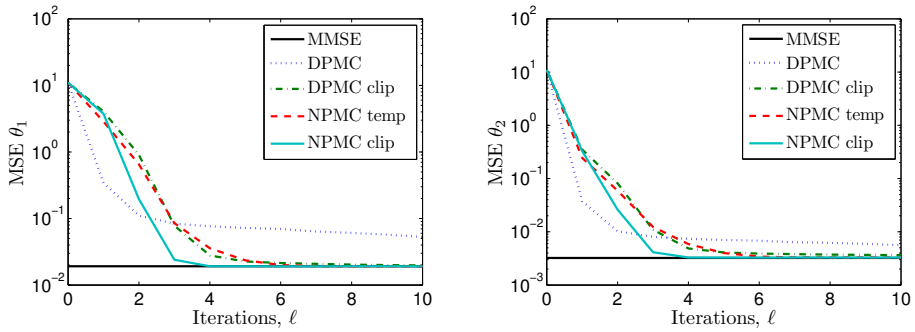

Figure 5.8: Evolution along the iterations of the average MSE for θ_1 (*left*) and θ_2 (*right*) for the DPMC, DPMC with clipping, NPMC with tempering and NPMC with clipping algorithms. The MMSE attainable for θ_1 and θ_2 are also represented, for reference, with solid black lines.

the DPMC with clipping and the proposed NPMC schemes outperform the original method in terms of MSE, reaching the MMSE in about 6 iterations.

However, the most outstanding difference in the performance of the analyzed algorithms is observed in the variance of the MSE. The final mean and standard deviation (std) values of the MSE for θ_1 and θ_2 at $\ell = L$ are shown in Table 5.1. The estimates provided by the DPMC scheme present a very high variance. On the contrary, the modified DPMC and the proposed NPMC schemes reach the MMSE, both in average and in standard deviation.

Assuming that the computation time for the DPMC method is 1, the DPMC with clipping takes 1.0006 time units (that is, only 0.06 % higher) and the NPMC schemes take 0.9565 and 0.9582 time units for the tempering and

Table 5.1: Mean and standard deviation (std) of the MSE of θ_1 and θ_2 at the last iteration $\ell = L$, for the studied PMC schemes. The MMSE (mean and std) corresponding to the true posterior $p(\boldsymbol{\theta}|\mathbf{y})$ is also shown for comparison. Note that all entries are multiplied by a factor of 10^3.

	MSE θ_1		MSE θ_2	
	mean $\times 10^3$	std $\times 10^3$	mean $\times 10^3$	std $\times 10^3$
DPMC	52.8	498.5	5.6	34.4
DPMC clip	19.7	14.1	3.6	2.4
NPMC temp	19.1	13.8	3.3	2.4
NPMC clip	19.1	13.8	3.3	2.4
MMSE	19.1	13.7	3.2	2.3

clipping schemes, respectively. This indicates that the compared methods have a very similar computational cost.

Selection of the M_T parameter

The NPMC algorithm with a clipping transformation has a single specific parameter M_T. Here we discuss on the choise of this value and illustrate the low sensitivity of the algorithm to its selection. We have thus performed $P = 10^4$ independent simulations of the NPMC method with a clipping transformation with $M = 200$ samples and $L = 10$ iterations, for values of the ratio M_T/M between 0 and 1. In Figure 5.9 we represent the average final NESS (*left*) and MSE of each parameter (*right*), versus M_T/M. It can be observed that the NPMC algorithm yields very good and similar results for values of M_T/M between 0.07 and 0.4, both in terms of NESS and MSE. For $M_T/M < 0.07$, which corresponds to $M_T = 14$ samples, the algorithm behaves similarly to the plain PMC algorithm with independent Gaussian proposals and standard IWs. The degeneracy of the IWs is not alleviated and the final NESS and MSE are poor. Note, however, that even setting M_T to a minimum possible value of $M_T = 2$ (which allows for the estimation of a bidimensional covariance matrix), the algorithm yields great improvement of the MSE w.r.t. to the prior MSE, and similar results to the minimum average MSE reached by the MH for the optimum value of σ (*right* plot in Figure 5.2). For values of M_T/M above 0.4, the distortion introduced by the clipping procedure hinders the convergence of the algorithm, yielding a low NESS and a high MSE. For values of M_T/M above 0.6, the NESS

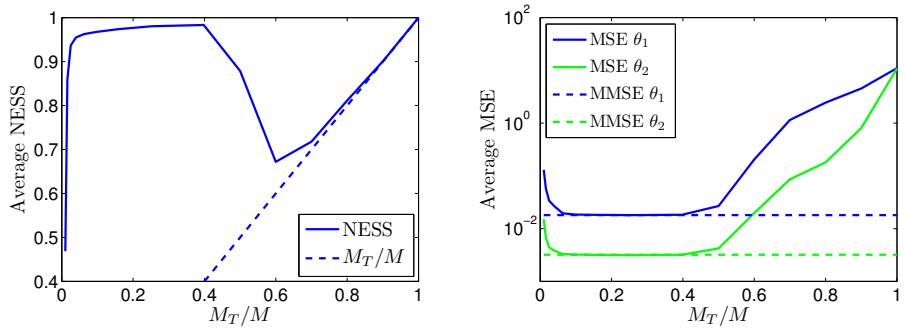

Figure 5.9: Average NESS and MSE of the NPMC method versus M_T/M.

artificially increases due to the clipping transformation only.

These simulations reveal that the NPMC algorithm presents low sensitivity to the selection of the M_T parameter, which makes the implementation of the algorithm fairly simple. Moreover, the distortion introduced by the clipping procedure is broadly offset by the benefits of the resulting tempering effect.

5.2 Nonlinear mixture PMC

In this section we compare the performance of the original MPMC algorithm proposed in [30] and reproduced in Table 2.10 with the proposed nonlinear MPMC method. We demonstrate how the computation of TIWs dramatically improves the performance of the original method, increasing its efficiency and robustness to numerical issues.

We consider a target density consisting of a Gaussian mixture in a 10-dimensional space. In particular, let us define

$$\begin{aligned}\pi(\boldsymbol{\theta}) = 0.35\mathcal{N}(\boldsymbol{\theta}; -2\mathbf{1}_{10}, 0.5\mathbf{I}_{10}) &+ 0.4\mathcal{N}(\boldsymbol{\theta}; 0.5\mathbf{1}_{10}, 0.25\mathbf{I}_{10}) \\ &+ 0.25\mathcal{N}(\boldsymbol{\theta}; 2\mathbf{1}_{10}, 0.5\mathbf{I}_{10}),\end{aligned}$$

where $\mathbf{1}_{10} = [1, \ldots, 1]^\top$ and $\mathbf{I}_{10}$ is the 10×10 identity matrix. For the NMPMC method, we use a clipping transformation and set $M_T = \sqrt{M}$.

5.2.1 IS vs nonlinear IS

If we restrict our attention to the first iteration of the PMC schemes, then we can carry out a comparison of the standard IS method (with conventional IWs) and the NIS technique (with TIWs). We consider an importance

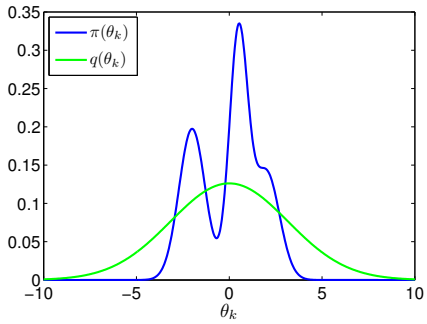

Figure 5.10: Marginal target, $\pi(\theta_k)$, and marginal proposal, $q(\theta_k)$, pdfs.

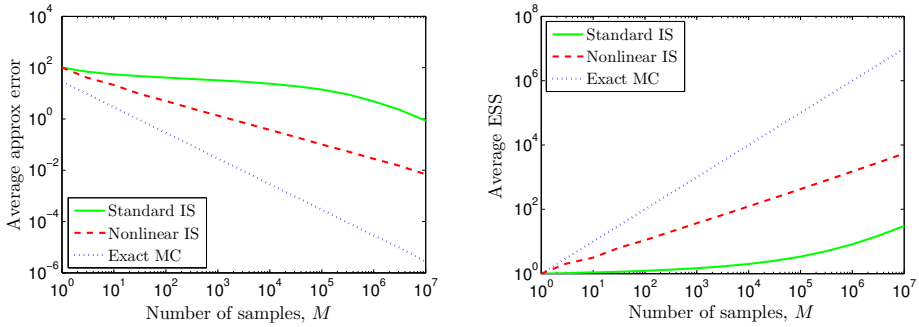

Figure 5.11: Average approximation error (*left*) and ESS (*right*) vs M with standard IS, NIS and exact Monte Carlo sampling (labeled as "exact MC").

function given by the 10-dimensional Gaussian pdf $q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \mathbf{0}_{10}, 10\mathbf{I}_{10})$, where $\mathbf{0}_{10} = [0, \ldots, 0]^\top$, which corresponds to a vague prior knowledge. Both the marginal target and proposal pdfs are represented in Figure 5.10.

We compute an estimate of the mean of $\pi(\boldsymbol{\theta})$, i.e., $\hat{\boldsymbol{\theta}} = \int \boldsymbol{\theta}\pi(\boldsymbol{\theta})d\boldsymbol{\theta}$, based on a set of M samples $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^M$ from $q(\boldsymbol{\theta})$, with standard IWs as $\hat{\boldsymbol{\theta}}^M = \sum_{i=1}^M w^{(i)}\boldsymbol{\theta}^{(i)}$ and with TIWs as $\bar{\boldsymbol{\theta}}^M = \sum_{i=1}^M \bar{w}^{(i)}\boldsymbol{\theta}^{(i)}$. In Figure 5.11 (*left*) we depict the approximation error obtained by IS ($\|\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}^M\|^2$) and NIS ($\|\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}^M\|^2$), as a function of the number of samples M, averaged over 10^4 independent simulation runs. The exact Monte Carlo error (approximating $\hat{\boldsymbol{\theta}}$ with samples generated from $\pi(\boldsymbol{\theta})$) is also depicted for comparison. It can be clearly observed that the approximation error obtained with NIS is far below the one obtained with standard IS. Thus, the number of samples needed with NIS to obtain a given approximation error is much lesser. Correspondingly, Figure 5.11 (*right*) shows that the average ESS obtained with standard IS increases more slowly than that with NIS.

5.2.2 MPMC vs nonlinear MPMC

Let us compare the performance of the original and nonlinear MPMC algorithms, in their plain and RB versions. We have performed 10^4 independent simulation runs of the following algorithms: MPMC, RB-MPMC, NMPMC and RB-NMPMC. In all the simulations, we have considered an initial proposal pdf $q_1(\boldsymbol{\theta})$ composed of $D = 5$ equally weighted Gaussian components with covariance matrix $10\mathbf{I}_{10}$ and random mean vectors. The number of iterations has been set to $L = 20$ and the number of samples per iteration to $M = 5000$. At each iteration of all PMC schemes we compute an approximation of the KLD between the target and the ℓ-th proposal by Monte Carlo simulation.

Figure 5.12 depicts the final NESS versus the final KLD in logarithmic scale obtained in each simulation run of the RB-MPMC (*left*) and the RB-NMPMC (*right*) algorithms, together with the corresponding histograms. We observe that most of the simulation runs of the RB-NMPMC algorithm end up with a low KLD (below 10^{-1}) and a NESS close to 1. Outcomes of this type correspond to exact matching of the final proposal to the target pdf and are classified into Group 1. Outcomes with a final KLD between 10^{-1} and $10^{0.5}$ belong to Group 2 and correspond to solutions in which some of the modes are grouped into one. The outcomes with a final KLD above $10^{0.5}$ belong to Group 3 and correspond to solutions where some of the modes are ignored. These threshold values are also represented in Figure 5.12 with solid red lines. Additionally, we define Group 4 containing simulation runs which ended with non-proper solutions or numerical errors.

In Table 5.2 the percentage of outcomes in each of the groups for each of the tested algorithms are shown. We observe that the NMPMC schemes clearly outperform the original MPMC method, which yields outcomes in Group 4 in most of the cases. The RB-NMPMC technique obtains $\approx 70\%$ of outcomes in Group 1 and presents no numerical errors.

Table 5.2: Percentage of simulation runs belonging to each group of the MPMC and NMPMC algorithms, in their plain and RB versions.

	Group 1	Group 2	Group 3	Group 4
MPMC	0 %	0 %	1 %	99 %
RB-MPMC	0 %	0.07 %	4.34 %	95.59 %
NMPMC	14.65 %	45.51 %	34.73 %	5.11 %
RB-NMPMC	**69.96 %**	**14.64 %**	**15.40 %**	**0 %**

In Figure 5.13 (*upper* row) the typical outcomes of the RB-MPMC method are represented, corresponding to Groups 2 (*left*), 3 (*center*) and 4 (*right*). In the *lower* row, the typical outcomes of nonlinear RB-MPMC are shown, corresponding to Groups 1 (*left*), 2 (*center*) and 3 (*right*). The RB-MPMC algorithm yields solutions belonging to Group 4 nearly always (*upper right* plot), while its nonlinear version reaches the exact solution in most of the cases (*lower left* plot).

Figure 5.14 shows the evolution along the iterations of the average KLD (*left*) and NESS (*right*), respectively, of the outcomes belonging to Groups 1, 2 and 3, attained with the RB NMPMC algorithm. Clear differences can be observed among the three groups. The outcomes of Group 1 reach, on average, a low final KLD and a high NESS. When some of the modes are ignored, the final NESS is also close to one, which can be misleading. However, both cases can be distinguished by the evolution of the NESS along the iterations, reaching a steady value faster in the second case.

5.3 Adaptive nonlinear mixture PMC

To illustrate the performance of the original MPMC method in [30] and the new adaptive NMPMC algorithm, we apply both schemes to the approximation of a 10-dimensional target pdf $\pi(\boldsymbol{\theta})$, by means of a mixture of either Gaussian or Student's t kernels.

Following [164], we consider a target pdf $\pi(\boldsymbol{\theta})$ constructed from a Gaussian pdf $\pi(\boldsymbol{\theta'}) = \mathcal{N}_{10}(\boldsymbol{\theta'}; \mathbf{0}, \boldsymbol{\Sigma})$ with covariance matrix $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, 1, \ldots, 1)$. The variable of interest $\boldsymbol{\theta}$ is constructed from the auxiliary variable $\boldsymbol{\theta'}$ by twisting the second coordinate according to $\theta_2 = \theta'_2 - \beta(\theta'^2_1 - \sigma_1^2)$ and keeping the rest of the variables unchanged, i.e.,

$$\boldsymbol{\theta} = [\theta'_1, \theta'_2 - \beta(\theta'^2_1 - \sigma_1^2), \theta'_3, \ldots, \theta'_{10}]^\top.$$

We assume that the twist parameter is $\beta = 0.03$ and $\sigma_1^2 = 100$. This transformation results in a banana-shaped density in the first two dimensions, which is represented in Figure 5.15 (*left*), together with a GMM sample approximation (*right*). This pdf is difficult to explore and provides a realistic model for many cosmological problems [91, 164].

We have applied the MPMC and the adaptive NMPMC methods to this problem with importance functions built as Gaussian and t mixtures. In all the simulations, we consider an initial proposal pdf consisting of $D_1 = 10$ components, with random mean vectors $\boldsymbol{\mu}_{1,d} \sim \mathcal{N}_{10}(\boldsymbol{\mu}_{1,d}; \mathbf{0}, \boldsymbol{\Sigma}_0/5)$, and a common covariance matrix $\boldsymbol{\Sigma}_{1,d} = \boldsymbol{\Sigma}_0$, where $\boldsymbol{\Sigma}_0 = \text{diag}(200, 50, 4, \ldots, 4)$.

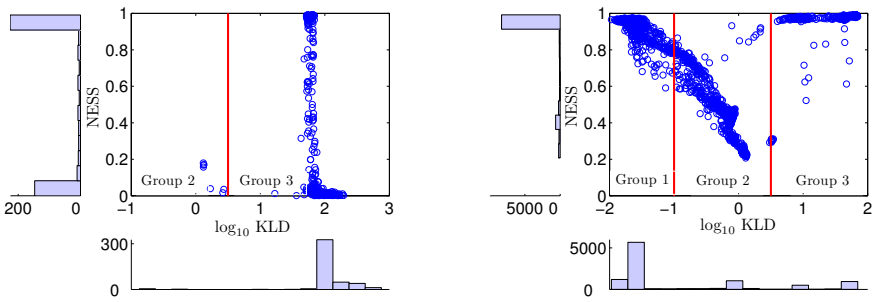

Figure 5.12: Final NESS vs final KLD obtained in each simulation run of the RB-MPMC (*left*) and RB-NMPMC (*right*) algorithms.

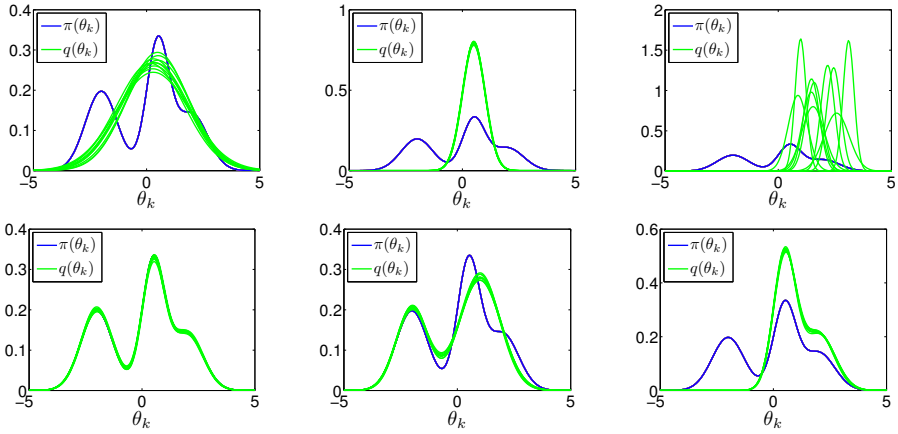

Figure 5.13: Typical outcomes of RB-MPMC (*upper* row) and RB-NMPMC (*lower* row). The target pdf $\pi(\boldsymbol{\theta}_k)$ is represented with blue lines while the last proposal pdfs $q_{L+1}(\theta_k)$, $k = 1, \ldots, K$, are represented with green lines.

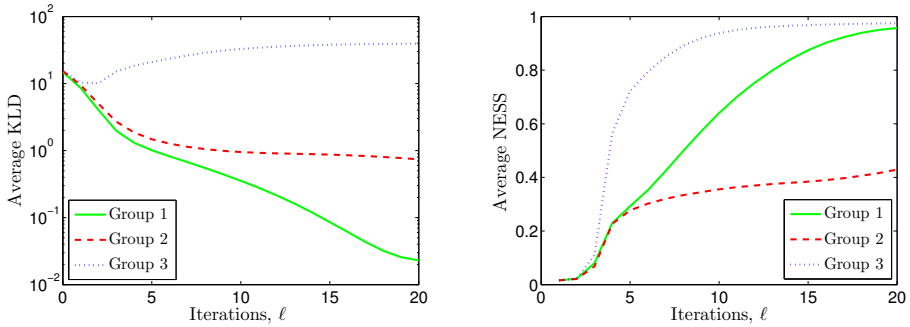

Figure 5.14: Evolution of the KLD (*left*) and NESS (*right*) along the iterations with the nonlinear RB-MPMC in each of the three groups.

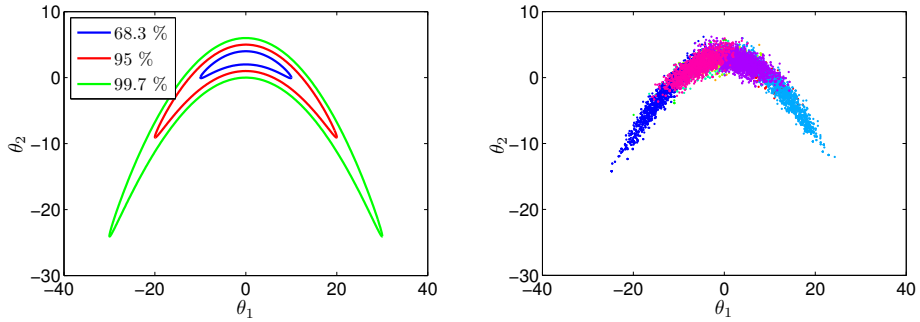

Figure 5.15: Contour plot of the marginal target pdf $\pi(\theta_1, \theta_2)$ (*left*) and a GMM approximation with $D = 7$ components and $M = 10^4$ samples (*right*).

In the case of t mixtures, the number of degrees of freedom has been set to $\nu_d = 9$, for $d = 1, \ldots, D_\ell$.

In each simulation run we have computed the NESS at all iterations as $M_\ell^{neff} = [M \sum_{i=1}^{M} (w_\ell^{(i)})^2]^{-1}$ and $\bar{M}_\ell^{neff} = [M \sum_{i=1}^{M} (\bar{w}_\ell^{(i)})^2]^{-1}$ for the MPMC and the adaptive NMPMC schemes, respectively.

As a measure of how well a set of samples $\{\boldsymbol{\theta}_\ell^{(i)}\}_{i=1}^M$ drawn from the mixture proposal pdf $q_\ell(\boldsymbol{\theta})$ represents the target density $\pi(\boldsymbol{\theta})$ we have computed the KLD between the corresponding Gaussian target pdf $\pi(\boldsymbol{\theta}')$ and the Gaussian approximation of the untwisted sample set $\{\boldsymbol{\theta}_\ell^{'(i)}\}_{i=1}^M$, obtained by the inverse transformation $\theta_{\ell,2}^{'(i)} = \theta_{\ell,2}^{(i)} + \beta[(\theta_{\ell,1}^{(i)})^2 - \sigma_1^2]$.

We have performed 10^4 independent simulation runs for each algorithm, both in the Gaussian and t cases. We have studied two settings with $L = 20$ iterations and a different number of samples per iteration, M.

5.3.1 Large sample size

The number of samples per iteration has been set to $M = 10^4$. The threshold parameter for the removal of a mixture component has been set to $\mu_{prn} = 0.002$. In the adaptive NMPMC scheme, the threshold parameter for the fusion of two components has been set to $\mu_{mrg} = 3$, and the clipping parameter to $M_T = 100$ samples.

In Figures 5.16 (*left*) and 5.17 (*left*) the evolution of the median KLD and mean NESS for the MPMC and the adaptive NMPMC algorithms are plotted, for the algorithms with Gaussian and t kernels. The median has been preferred to the mean because of its robustness against outliers. The proposed scheme obtains a lower KLD and a higher NESS with both

families of mixture. The Gaussian mixture provides better results, in terms of KLD and NESS, than the t mixture for both algorithms. This occurs because samples drawn from the tails of the t components are not usually representative and obtain low IWs. The evolution of the number of components D_ℓ, shown in Figure 5.18 (*left*), is similar in all the schemes, converging in average to a value in the interval [6,7].

In Table 5.3, various statistics on the values of KLD, NESS and D_ℓ after the last iteration $\ell = L$ are displayed. It can be observed that the original MPMC schemes present an extremely high KLD variance and also a higher variance of NESS and D_ℓ than the proposed techniques.

5.3.2 Reduced sample size

In this case, the number of samples per iteration has been set to $M = 2000$. The threshold parameters have been set to $\mu_{prn} = 0.01$ and $\mu_{mrg} = 2$, for prunning and merging components, respectively, and $M_T = 100$ samples.

Figures 5.16 (*right*), 5.17 (*right*) and 5.18 (*right*) display the results obtained in this setting. In this scenario, the MPMC method performs poorly with both mixture families, obtaining an increasing KLD, a NESS close to 0 and a mean D_L close to 1. On the contrary, the proposed NMPMC algorithm performs similarly to the $M = 10^4$ case, with a slightly higher KLD due to the fact that, with a lower number of samples, the tails of the target pdf are less accurately represented. The number of components D_ℓ still attains a similar final value and the NESS converges to a high value as well. In Table 5.4 various statistics on the final values of the KLD, NESS and D_ℓ for the NMPMC are also displayed for comparison. It can be seen that the proposed scheme presents stable results also in this scenario.

5.4 Summary and conclusions

In this chapter we have conducted a number of computer simulation experiments that demonstrate the performance of the proposed NPMC algorithm in simple numerical examples.

We have considered a simple GMM to illustrate the degeneracy problem, which can often arise even in low dimensional problems. We have compared the performance of the NPMC method, with clipping and tempering transformations, to other standard techniques, such as the MH and the DPMC algorithm. The best results have been obtained with the NPMC method with a clipping transformation. This algorithm is barely sensitive to the selection of the parameter M_T, which can be selected between $0.1M$ and

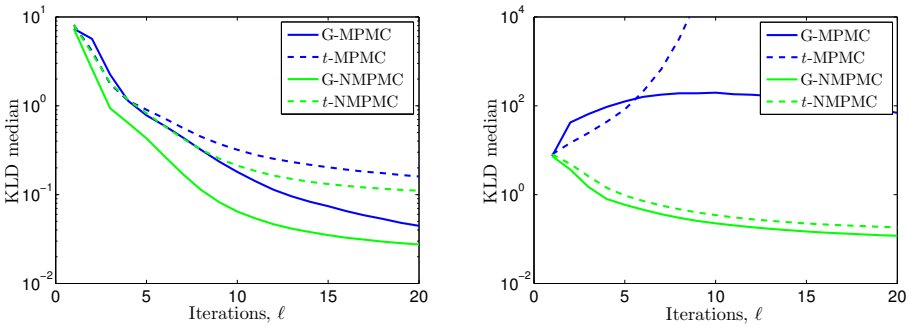

Figure 5.16: Median KLD along the iterations with $M = 10^4$ (*left*) and $M = 2000$ (*right*), with Gaussian and t mixtures.

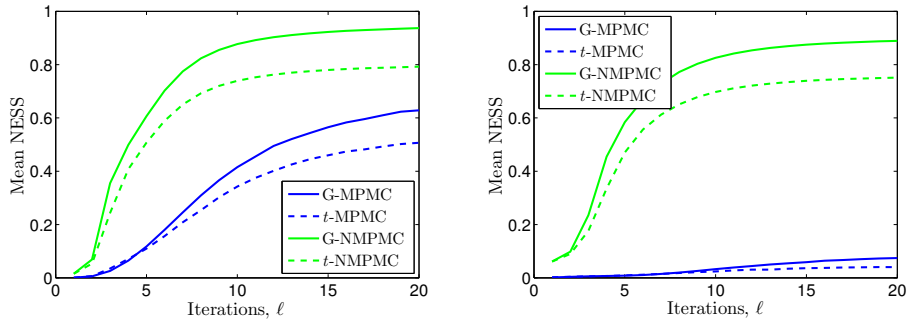

Figure 5.17: Mean NESS along the iterations with $M = 10^4$ (*left*) and $M = 2000$ (*right*).

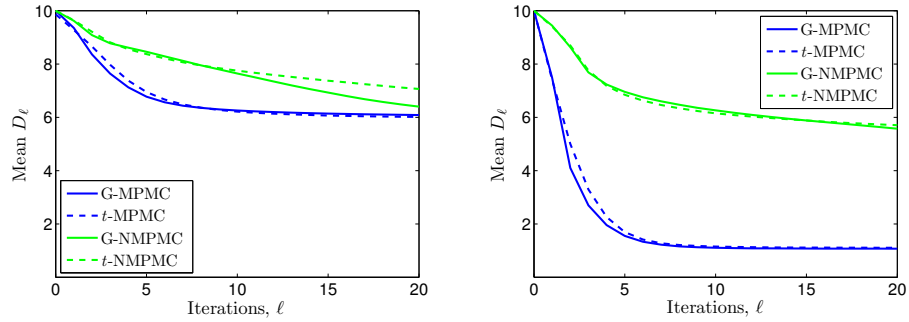

Figure 5.18: Mean number of mixture components D_ℓ along the iterations with $M = 10^4$ (*left*) and $M = 2000$ (*right*).

Table 5.3: Median, mean and standard deviation of the KLD, NESS and D_ℓ, for MPMC and NMPMC with $M = 10^4$ and $\ell = L$.

	G-MPMC	t-MPMC	G-NMPMC	t-NMPMC
Med KLD	0.0445	0.1601	0.0275	0.1106
Mean KLD	$1.63 \cdot 10^5$	$1.06 \cdot 10^{26}$	0.0307	0.1147
Std KLD	$1.47 \cdot 10^7$	$5.76 \cdot 10^{27}$	0.0139	0.0254
Mean NESS	0.6286	0.5069	0.9370	0.7923
Std NESS	0.2699	0.2492	0.0176	0.0148
Mean D_L	6.084	6.013	6.401	7.065
Std D_L	2.242	2.501	1.167	1.310

Table 5.4: Median, mean and standard deviation of the KLD, NESS and D_ℓ, for NMPMC with $M = 2000$ and $\ell = L$.

	G-NMPMC	t-NMPMC
Med KLD	0.1182	0.1832
Mean KLD	0.1287	0.1949
Std KLD	0.0514	0.0627
Mean NESS	0.8892	0.7512
Std NESS	0.0153	0.0142
Mean D_L	5.573	5.705
Std D_L	1.276	1.295

$0.4M$ in many applications. Our simulations reveal that the ESS, both for MCMC and PMC, is a useful indicator of the performance of the algorithm. Thus, a high ESS suggests that the obtained sample is representative of the target distribution, and often corresponds to low MSE estimates.

We have also performed numerical simulations of the MPMC algorithm proposed in [30] and its nonlinear version, which additionally updates the number of mixture components at each iteration. We have used a 10 dimensional GMM and a banana-shaped target density to assess the performance of both algorithms, with Gaussian and Student's t proposals and with a different number of samples (hence, with a different computational effort). We present numerical results that show that the proposed scheme clearly outperforms the original one. We also show that the Gaussian mixture should be preferred for this problem.

Chapter 6

Bayesian inference in stochastic kinetic models

In this chapter we numerically assess the performance of the proposed particle NPMC algorithm for Bayesian inference in state-space models. As a practical application, we have chosen the challenging problem of approximating the joint posterior distribution of the parameters $\boldsymbol{\theta}$ and the hidden states $\mathbf{x}$ in stochastic kinetic models (SKMs). In Section 6.1 we introduce the basics of SKMs. In Section 6.2 we describe the usual solutions to this problem from a Bayesian approach. In Section 6.3 we present simulation results for the PNPMC algorithm when applied to a simple SKM known as the predator-prey model, consisting of two interacting species related by three reaction equations with associated unknown rates [158]. In Section 6.4 we numerically compare the performance of PMCMC and PNPMC schemes when applied to a challenging prokaryotic autoregulatory model [71, 72, 161], in two scenarios of different dimension and with two different observation models. Finally, Section 6.5 is devoted to the conclusions of this chapter.

6.1 Stochastic kinetic models

Stochastic kinetic models (SKMs) are continuous-time jump processes modeling the interactions among molecules, or species, that take place in chemical reaction networks of biochemical and cellular systems, according to a set of unknown rate parameters [161].

Consider a biochemical reaction network that describes the time evolution of the population of V chemical species $x_1, \ldots, x_V$ related by

means of K reactions $r_1, \ldots, r_K$

$$\begin{aligned}
r_1: &\quad p_{11}x_1 + p_{12}x_2 + \ldots + p_{1V}x_V \xrightarrow{c_1} q_{11}x_1 + q_{12}x_2 + \ldots + q_{1V}x_V, \\
r_2: &\quad p_{21}x_1 + p_{22}x_2 + \ldots + p_{2V}x_V \xrightarrow{c_2} q_{21}x_1 + q_{22}x_2 + \ldots + q_{2V}x_V, \\
&\quad \vdots \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
r_K: &\quad p_{K1}x_1 + p_{K2}x_2 + \ldots + p_{KV}x_V \xrightarrow{c_K} q_{K1}x_1 + q_{K2}x_2 + \ldots + q_{KV}x_V,
\end{aligned}$$

where p_{kv} and q_{kv}, $k = 1, \ldots, K$, $v = 1, \ldots, V$, denote the reactant and the product coefficients, respectively; and $c_k > 0$, $k = 1, \ldots, K$, are the random constant rate parameters. A matrix $\mathbf{P}$ of size $K \times V$ contains the reactant coefficients p_{kv} and, similarly, $\mathbf{Q}$ contains the product coefficients q_{kv}. The stoichiometry matrix of size $V \times K$ is defined as $\mathbf{S} = (\mathbf{Q} - \mathbf{P})^\top$. The vector $\mathbf{c} = [c_1, \ldots, c_K]^\top$ contains the rate parameters.

Let $x_v(t)$, $v = 1, \ldots, V$, denote the nonnegative, integer population of species x_v at time t, and let $\mathbf{x}(t) = [x_1(t), \ldots, x_V(t)]^\top$ denote the state of the system at this time instant. Let $\mathbf{x}_n = [x_{1,n}, \ldots, x_{V,n}]^\top$ denote the state of the system at discrete time instants $t = n\Delta$, $n = 1, \ldots, N$, where $x_{v,n} = x_v(n\Delta)$ and Δ denotes a time-discretization period. We denote by $\mathbf{x}$ the vector containing the population of each species at N consecutive discrete time instants, i.e., $\mathbf{x} = [\mathbf{x}_1^\top, \ldots, \mathbf{x}_N^\top]^\top$.

The k-th reaction takes place stochastically according to its instantaneous rate or hazard function [161]

$$h_k(t) = c_k \prod_{v=1}^{V} \binom{x_v(t)}{p_{kv}}, \quad k = 1, \ldots, K,$$

where the product of binomial coefficients represents the number of combinations in which the k-th reaction can occur, as a function of the population of each reactant species x_v. We additionally define the vector $\mathbf{h}(t) = [h_1(t), \ldots, h_K(t)]^\top$. The waiting time to the next reaction is exponentially distributed with parameter $h_0(t) = \sum_{k=1}^{K} h_k(t)$, and the probability of each reaction type is given by $h_k(t)/h_0(t)$.

The Gillespie algorithm [69], which is displayed in Table 6.1, allows to generate exact forward simulations of arbitrary SKMs, by drawing samples from the transition density $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{c})$, $n = 1, \ldots, N$, given a set of rate parameters $\mathbf{c}$ and an initial population $\mathbf{x}_0$. The algorithm can be run up to a number of reactions or for a given time interval T.

Table 6.1: Gillespie algorithm [69].

Initialization:

1. Set $t = 0$, $r = 0$ and select a time interval length T and an initial population vector $\mathbf{x}(0)$.

Iterations:

1. Compute the instantaneous rates $h_k(t)$ and $h_0(t) = \sum_{k=1}^{K} h_k(t)$. The probability of reaction k is $p_k(t) = h_k(t)/h_0(t)$.
2. Generate two random numbers $s_1, s_2 \sim \mathcal{U}[0, 1]$.
3. Compute the waiting time to the next reaction as $\tau(t) = (1/h_0(t)) \ln(1/s_1)$.
4. Select the reaction type according to s_2 and the probability of each reaction $p_k(t)$.
5. Update the time index $t = t+\tau(t)$, and the reaction index $r = r+1$.
6. Adjust the populations of the species $\mathbf{x}(t)$ according to the reaction occurred.
7. If $t < T$ go to step 1.

6.2 Bayesian inference for SKMs

Following [161] we consider the log-transformed rate parameters $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_K]^\top$, where $\theta_k = \log(c_k)$, $k = 1, \ldots, K$, with prior pdf $p(\boldsymbol{\theta})$. The prior pdf of the initial population vector $\mathbf{x}_0$ is denoted $p(\mathbf{x}_0)$. We assume that a linear combination of the populations of a subset of species is observed at discrete time instants corrupted by Gaussian noise, i.e.,

$$\mathbf{y}_n = \mathbf{M}\mathbf{x}_n + \mathbf{w}_n, \quad n = 1, \ldots, N, \tag{6.1}$$

where $\mathbf{M}$ is the observation matrix with dimensions $D \times V$ and $\mathbf{w}_n \sim \mathcal{N}_D(\mathbf{w}_n; \mathbf{0}, \boldsymbol{\Lambda})$ is a multivariate Gaussian noise component with zero mean and covariance matrix $\boldsymbol{\Lambda}$. We denote the complete observation vector with dimension $DN \times 1$ as $\mathbf{y} = [\mathbf{y}_1^\top, \ldots, \mathbf{y}_N^\top]^\top$.

The dynamical behavior of an arbitrary SKM can be described in terms of a state-space model as in equation (2.3), by means of a transition pdf $p(\mathbf{x}_n|\mathbf{x}_{n-1},\boldsymbol{\theta})$ and a likelihood function $p(\mathbf{y}_n|\mathbf{x}_n)$. In this work we aim to obtain a Monte Carlo approximation of the full joint posterior pdf of the log-rate parameters $\boldsymbol{\theta}$ and the populations $\mathbf{x}$, given by equation (2.8).

We are also interested in computing approximations of the posterior marginals of the rate parameters $p(\boldsymbol{\theta}|\mathbf{y}) = \int p(\boldsymbol{\theta},\mathbf{x}|\mathbf{y})d\mathbf{x}$ and the species populations $p(\mathbf{x}|\mathbf{y}) = \int p(\boldsymbol{\theta},\mathbf{x}|\mathbf{y})d\boldsymbol{\theta}$ as well as their moments (e.g., the posterior mean), which are of the form

$$\begin{aligned} E_{p(\boldsymbol{\theta}|\mathbf{y})}[f(\boldsymbol{\theta})] &= \int f(\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}, \quad \text{and} \\ E_{p(\mathbf{x}|\mathbf{y})}[f(\mathbf{x})] &= \int f(\mathbf{x})p(\mathbf{x}|\mathbf{y})d\mathbf{x}, \end{aligned}$$

respectively, where f is a real, integrable function. The likelihood function $p(\mathbf{y}|\boldsymbol{\theta})$ is given by equation (2.9) and cannot be evaluated exactly. However, it can be approximated via a standard PF as described in Section 2.4.1.

This inference problem has been traditionally addressed using MCMC methods, and IS based schemes have been avoided due to their inefficiency in high dimension [161]. In [26] various MCMC algorithms are evaluated in data-poor scenarios. In [72] the PMCMC algorithm described in Section 2.5.2 was applied to this problem. This method is, to the best of our knowledge, the most powerful, yet computationally expensive, method provided so far for this kind of applications.

Bayesian inference based on exact stochastic simulations from $p(\mathbf{x}_n|\mathbf{x}_{n-1},\boldsymbol{\theta})$ generated via the Gillespie algorithm often becomes practically intractable even for models of modest complexity [71]. Thus, it is very common to resort to a continuous approximation of the underlying stochastic process, which is known as the diffusion approximation [161]. This approximation is known to be poor in low concentration scenarios, and thus should be avoided for models involving species with a very low population. Alternatively, the authors of [128] propose an approximation of the likelihood based on the moment closure approximation of the underlying stochastic process.

We propose to apply the PNPMC algorithm described in Section 3.5 for the approximation of the joint posterior distribution of the rate parameters and the populations of all species, provided a set of discrete, noisy observations is available. We have applied the proposed algorithm to two SKMs of different complexity: the simple predator-prey model [158, 161] and the more challenging prokaryotic autoregulatory model [72, 161].

6.3 Predator-prey model

The Lotka-Volterra, or predator-prey, model is a simple SKM that describes the time evolution of the populations of two species $x_1(t)$ (prey) and $x_2(t)$ (predator), $t \in \mathbb{R}$, by means of $K = 3$ reaction equations [158]

$$\begin{array}{rl} x_1 \xrightarrow{c_1} 2x_1 & \text{prey reproduction} \\ x_1 + x_2 \xrightarrow{c_2} 2x_2 & \text{predator reproduction} \\ x_2 \xrightarrow{c_3} \emptyset & \text{predator death} \end{array}$$

where $\mathbf{c} = [c_1, c_2, c_3]^\top$ is the vector of constant (yet random) rate parameters $c_k > 0$, $k = 1, 2, 3$. Let $\mathbf{x}_n = [x_{1,n}, x_{2,n}]^\top$ denote the state of the system at time instant $t = n\Delta$, $n = 1, \ldots, N$.

We consider two different observation scenarios. In the complete observation (CO) scenario we assume that both species x_1 and x_2 are observed at regular time intervals and corrupted by Gaussian noise, i.e., $\mathbf{y}_n = \mathbf{x}_n + \mathbf{w}_n$, where $\mathbf{w}_n \sim \mathcal{N}_2(\mathbf{w}_n; \mathbf{0}, \mathbf{\Lambda})$, $n = 1, \ldots, N$. The complete vector of observations $\mathbf{y}$ has dimension $2N \times 1$.

In the partial observation (PO) scenario only x_1 is observed at discrete time instants and also contaminated by Gaussian noise, i.e., $y_n = x_{1,n} + w_n$, where $w_n \sim \mathcal{N}(w_n; 0, \sigma^2)$, $n = 1, \ldots, N$. In the PO case, the vector of scalar observations with dimension $N \times 1$ is constructed as $\mathbf{y} = [y_1, \ldots, y_N]^\top$.

In this section we present simulation results that illustrate the performance of the PNPMC algorithm applied to the approximation of the posterior distribution of the log-rate parameters $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3]^\top$ (where $\theta_k = \log(c_k)$, $k = 1, 2, 3$), with density $p(\boldsymbol{\theta}|\mathbf{y})$, in the CO and PO scenarios.

6.3.1 Simulation setup

Following [26, 72], the true vector of rate parameters which we aim to estimate has been set to $\mathbf{c}_* = [0.5, 0.0025, 0.3]^\top$, which yields

$$\boldsymbol{\theta}_* = [-0.69, -5.99, -1.20]^\top.$$

The initial populations and the number of observations have been set to $\mathbf{x}_0 = [100, 100]^\top$ and $N = 50$, respectively. The discretization period is $\Delta = 1$ and the noise variance is $\sigma^2 = 100$ (and assumed to be known). Uniform priors $\mathcal{U}(\theta_k; -7, 2)$ are taken for each $\theta_k = \log(c_k)$, and independent Poisson priors $p(x_{l,0}) = \mathcal{P}(x_{l,0}; \lambda_l)$ are considered for the initial populations $x_{l,0}$, with parameters set to the true values, that is, $\lambda_l = x_{l,0}$, $l = 1, 2$.

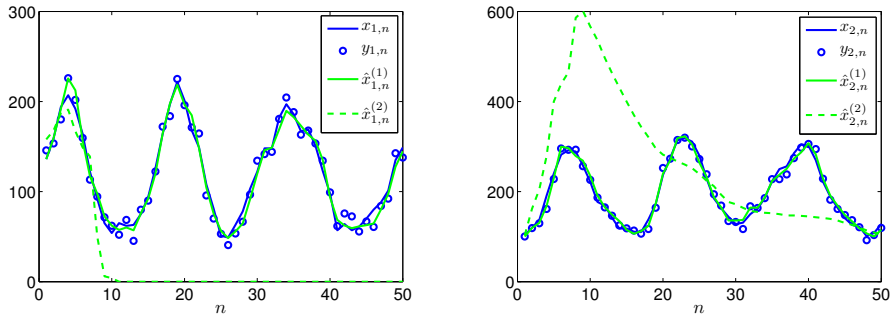

Figure 6.1: Observations, true and estimated populations of preys (*left*) and predators (*right*) obtained via a PF with two different parameter vectors $\boldsymbol{\theta}^{(1)} = \boldsymbol{\theta}_*$ and $\boldsymbol{\theta}^{(2)} = [-0.12, -5.51, -3.11]^\top$, in the CO scenario.

The number of particles of the PF used to compute the likelihood approximation $\hat{p}^J(\mathbf{y}|\boldsymbol{\theta}^{(i)})$ has been set to $J = 100$. Increasing J improves the performance only slightly, at the expense of a significant increase of the computational cost (this is coherent with the results, e.g., in [161, 72], where the same value of J is selected).

In Figure 6.1 we illustrate the performance of the PF at the core of the PNPCM algorithm. The true and estimated populations, together with the observations of preys (*left*) and predators (*right*) are represented for two different parameter vectors $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$. The populations of both species have been approximated as the posterior mean of $\mathbf{x}$ given the parameter vector $\boldsymbol{\theta}^{(i)}$, obtained as in equation (2.16) via a PF.

We have also computed the marginal likelihood approximation $\hat{p}^J(\mathbf{y}|\boldsymbol{\theta}^{(i)})$ using the PF for both parameter vectors $\boldsymbol{\theta}^{(i)}$, according to equations (2.17) and (2.18). In the first case the parameters have been set to the true values $\boldsymbol{\theta}^{(1)} = \boldsymbol{\theta}_*$ and the obtained log-likelihood was $\log(\hat{p}^J(\mathbf{y}|\boldsymbol{\theta}^{(1)})) = -413$. In the second case, the parameter vector $\boldsymbol{\theta}^{(2)} = [-0.12, -5.51, -3.11]^\top$ has been drawn from the prior pdf $p(\boldsymbol{\theta})$ and the obtained log-likelihood was $\log(\hat{p}^J(\mathbf{y}|\boldsymbol{\theta}^{(2)})) = -9940$.

In Figure 6.1 it can be observed that, when the parameters are set to the true values, both population trajectories are much better estimated and the obtained likelihood takes a much higher value than when the parameters are poorly chosen. Thus, the parameter vector $\boldsymbol{\theta}^{(1)}$ would attain a much higher IW than $\boldsymbol{\theta}^{(2)}$, if drawn by the PNPMC algorithm.

Despite the low dimension of this problem ($K = 3$), the IWs of the PMC scheme present severe degeneracy (as can be seen from the extreme difference

in the likelihood obtained for $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$ in the previous example), partly due to the likelihood approximation, which introduces additional variations to the IWs. Thus, the original PMC scheme with standard IWs does not work in this scenario. The PNPMC scheme with tempering also performs poorly compared to the method with clipping. Given the extreme variations of the IWs, it is not straightforward to select a priori a tempering sequence γ_ℓ which provides a sufficient ESS at all iterations. For this reason, we have focused on the PNPMC scheme with clipping, which computes TIWs at all iterations and guarantees a baseline ESS.

6.3.2 Results

We have performed $P = 100$ independent simulation runs of the PNPMC with clipping in the CO and the PO scenarios, with the same initial populations $\mathbf{x}_0$ and different (independent) population and observation vectors. Both in the CO and the PO cases, the same true population trajectories have been used, i.e., only the observations differ. The number of iterations has been set to $L = 10$, the number of samples per iteration is $M = 10^3$ and the clipping parameter is $M_T = 100$.

In the CO scenario, 5 simulation runs ended with a numerical error or with a final NESS value close to M_T/M, and were repeated, for the same observation vectors, with $M = 2000$ and $M_T = 200$. Numerical errors may occur when very few samples $\boldsymbol{\theta}_\ell^{(i)}$ attain a significant likelihood, specially at the first iteration. The NESS allows to detect whether the algorithm converges properly, when its value increases along the iterations beyond M_T/M. Thus, the average number of samples per iteration required in the CO case was $M = 1050$. On the contrary, in the PO case all the simulation runs ended satisfactorily with $M = 1000$.

In Figure 6.2 (*left*) the final values of the MSE ($MSE_{L,k}$) averaged over the parameters θ_k, $k = 1, 2, 3$, versus the final NESS $\bar{M}_L^{neff}$ obtained at each simulation run are depicted, in the CO (green circles) and the PO (blue squares) scenarios, together with the histogram of each variable. It can be observed that in the CO scenario a lower MSE is attained compared to the PO scenario, given the larger amount of data available. However, the NESS is also lower in the first case, which indicates more degeneracy of the IWs, again due to the larger amount of data. The required number of samples is larger in this case, being more computationally demanding and more sensitive to numerical issues. The big circle and square represent two particular simulation runs which attained a final MSE close to the global average value in the CO and PO scenarios, respectively.

Figure 6.2 (*right*) depicts the final estimate of the marginal posteriors $p(\theta_k|\mathbf{y})$ for the simulation runs represented as a big circle (CO) and square (PO) in Figure 6.2 (*left*). We have built a Gaussian approximation of the marginal posteriors, namely $\hat{p}^{M,J}(\theta_k|\mathbf{y}) = \mathcal{N}(\theta_k; \mu_k, \sigma_k^2)$, where μ_k and σ_k are the k-th mean and standard deviation components of $\boldsymbol{\mu}_{L+1}$ and $\boldsymbol{\Sigma}_{L+1}$, computed as in equations (3.2) and (3.3), respectively. It can be observed that the proposed algorithm successfully identifies the log-rate parameters both in the CO and the PO scenarios, and is robust to degeneracy problems that arise due to a large number of observations (specially in the CO case) and due to the approximation of the likelihood.

Table 6.2 shows the μ_k and σ_k parameters, $k = 1, 2, 3$, and the MSE, for the average simulation runs represented in Figure 6.2 (*left*), and whose estimates $\hat{p}^{M,J}(\theta_k|\mathbf{y})$ are depicted in Figure 6.2 (*right*), in both scenarios.

Figure 6.3 (*left*) shows the evolution of the average NESS in the CO and PO case. Both the NESS computed with standard IWs (M_ℓ^{neff}) and TIWs ($\bar{M}_\ell^{neff}$) are represented, with dashed and solid lines, respectively. Both M_ℓ^{neff} and $\bar{M}_\ell^{neff}$ increase beyond the effect of the clipping procedure, which indicates that the algorithm is able to generate more representative samples as it converges. Figure 6.3 (*right*) shows the evolution of the average MSE in the CO and PO case. The value of the MSE at $\ell = 0$ corresponds to the MSE obtained from the prior pdf. It can be seen that the MSE smoothly decreases up to a low final value, in just a few iterations.

The results presented here for the CO scenario can be compared, with some caution, to those obtained in [72] with a PMCMC scheme. The simulation setup is very similar, but the synthetic datasets employed here ($P = 100$ independent realizations of $\mathbf{y}$) and in [72] are different, as well as the prior describing the initial populations. Our simulations show that nearly equivalent results can been attained with the PNPMC method, which involves a considerably lower computational cost. Note that the effort demanded to process one PNPMC sample $\boldsymbol{\theta}_\ell^{(i)}$ is approximately equivalent to that of a single PMCMC iteration. In [72] 5×10^5 PMCMC iterations were run to compute solutions for this problem, while the PNPMC scheme has only required 10^4 samples overall (taking into account *all* the iterations), reducing the computational cost by a factor of 50 for a similar performance.

6.4 Prokaryotic autoregulatory model

In this section, we compare the performance of the PMCMC and the PNPMC methods when applied to the problem of approximating the

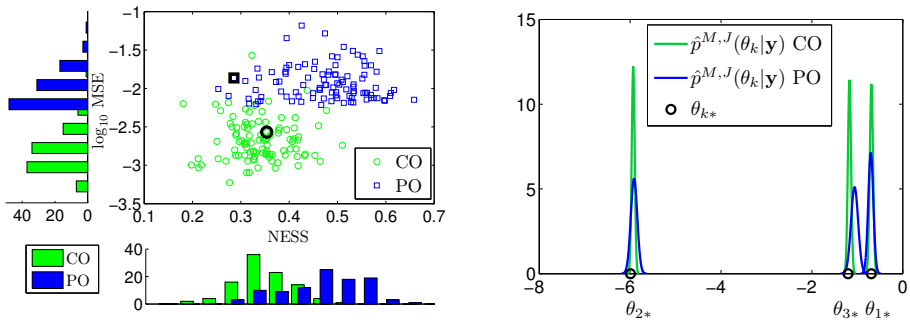

Figure 6.2: *Left*: Final MSE in logarithmic scale versus the final NESS in the CO and the PO scenario, together with the corresponding histograms. The big markers represent two average simulation runs. *Right*: Marginal estimated posteriors $\hat{p}^{M,J}(\theta_k|\mathbf{y})$ and true values θ_{k*}, $k = 1, 2, 3$, of the simulation runs represented as big markers in the *left* plot.

Table 6.2: Parameters and MSE of the Gaussian approximations $\hat{p}^{M,J}(\theta_k|\mathbf{y})$ for the average simulation run in the CO and PO case.

		μ_k	σ_k	MSE
$\hat{p}^{M,J}(\theta_1\|\mathbf{y})$	CO	-0.690	0.036	1.29×10^{-3}
	PO	-0.704	0.056	3.25×10^{-3}
$\hat{p}^{M,J}(\theta_2\|\mathbf{y})$	CO	-5.932	0.033	4.62×10^{-3}
	PO	-5.913	0.071	11.16×10^{-3}
$\hat{p}^{M,J}(\theta_3\|\mathbf{y})$	CO	-1.173	0.035	2.19×10^{-3}
	PO	-1.061	0.078	26.65×10^{-3}

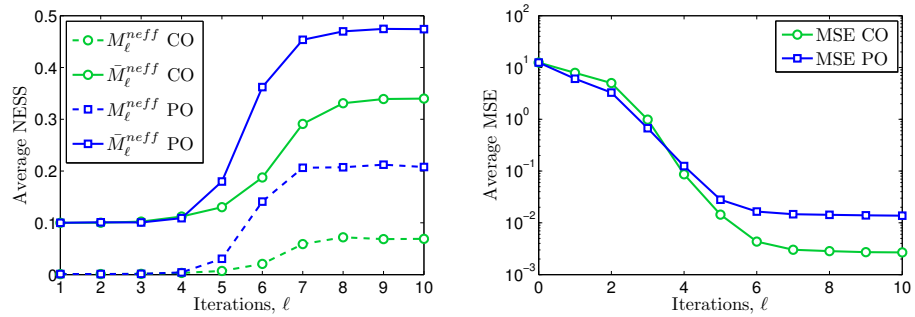

Figure 6.3: Average NESS (*left*) and MSE (*right*) along the iterations in the CO and PO scenarios. In the *left* plot M_ℓ^{neff} (dashed lines) are computed from standard IWs and $\bar{M}_\ell^{neff}$ (solid lines) are computed from TIWs.

posterior distributions of the log-rate parameters $p(\boldsymbol{\theta}|\mathbf{y})$ and the populations $p(\mathbf{x}|\mathbf{y})$ in a simplified prokaryotic autoregulatory model, given some observed data $\mathbf{y}$. This problem has been introduced in [71], and further analyzed in [72, 161]. This prokaryotic model is minimal in terms of the level of details included and offers a simplistic view of the mechanisms involved in gene autoregulation. However, it contains many of the interesting features of an auto-regulatory feedback network and does provide sufficient detail to capture the network dynamics. This model is significantly more complex than the predator-prey model studied in [18], due to the larger dimension of the parameter vector $\boldsymbol{\theta}$, the state $\mathbf{x}$ and the observations $\mathbf{y}$.

6.4.1 Prokaryotic autoregulation

The prokaryotic autoregulatory model is a SKM that involves $V = 5$ chemical species and $K = 8$ reaction equations, $r_1, \ldots, r_K$, given by

$$\begin{array}{ll} r_1 : x_{DNA} + x_{P_2} \xrightarrow{c_1} x_{DNA\cdot P_2}, & r_5 : 2x_P \xrightarrow{c_5} x_{P_2}, \\ r_2 : x_{DNA\cdot P_2} \xrightarrow{c_2} x_{DNA} + x_{P_2}, & r_6 : x_{P_2} \xrightarrow{c_6} 2x_P, \\ r_3 : x_{DNA} \xrightarrow{c_3} x_{DNA} + x_{RNA}, & r_7 : x_{RNA} \xrightarrow{c_7} 0, \\ r_4 : x_{RNA} \xrightarrow{c_4} x_{RNA} + x_P, & r_8 : x_P \xrightarrow{c_8} 0. \end{array}$$

We construct the V-dimensional vector containing the population of each species at time instant t as

$$\mathbf{x}(t) = [x_{RNA}(t), x_P(t), x_{P_2}(t), x_{DNA\cdot P_2}(t), x_{DNA}(t)]^\top.$$

Thus, we obtain a stoichiometry matrix of the form

$$\mathbf{S} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 2 & 0 & -1 \\ -1 & 1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and the hazard vector is given by

$$\begin{aligned} \mathbf{h}(t) = \; [\; & c_1 x_{DNA} x_{P_2}, c_2 x_{DNA\cdot P_2}, c_3 x_{DNA}, c_4 x_{RNA}, \\ & c_5 \frac{x_P(x_P - 1)}{2}, c_6 x_{P_2}, c_7 x_{RNA}, c_8 x_P]^\top, \end{aligned} \tag{6.2}$$

where the time dependence of the population of each species is omitted for simplicity of notation.

This model contains a conservation law given by the relation $x_{DNA\cdot P_2} + x_{DNA} = C$, where C is the number of copies of this gene in the genome. We could use this relation to remove $x_{DNA\cdot P_2}$ from the model, replacing any occurrences of the latter in the hazard function with $C - x_{DNA}$, but in this work we abide by the notation in equation (6.2). Further details of this model can be found in [161].

6.4.2 Simulation setup

We have selected most of the simulation parameters following [72]. The true vector of rate parameters which we aim to estimate has been set to

$$\mathbf{c}_* = [0.1, 0.7, 0.35, 0.2, 0.1, 0.9, 0.3, 0.1]^\top,$$

which yields log-transformed rate parameters

$$\boldsymbol{\theta}_* = -[2.30, 0.36, 1.05, 1.61, 2.30, 0.10, 1.20, 2.30]^\top.$$

The initial populations and the conservation constant have been set to $\mathbf{x}_0 = [x_1(0), \ldots, x_V(0)] = [8, 8, 8, 5, 5]^\top$ and $C = 10$, respectively. The time discretization period is $\Delta = 1$ and the Gaussian noise covariance matrix is $\boldsymbol{\Lambda} = \sigma^2\mathbf{I}$, with $\sigma^2 = 4$ (and assumed to be known). The number of particles for the PF approximation $\hat{p}^J(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$, has been set to $J = 100$ for all the simulations.

Independent uniform priors $\mathcal{U}(\theta_k; -7, 2)$ are taken for each $\theta_k = \log(c_k)$. Opposite to [72], the initial populations $\mathbf{x}_0$ are assumed unknown for the inference algorithm and we consider independent Poisson priors $p(x_v(0)) = \mathcal{P}(x_v(0); \lambda_v)$, with parameters set to the true initial populations, that is, $\lambda_v = x_v(0)$, $v = 1, \ldots, V$.

We again consider two different observation scenarios. In the CO scenario we assume that all species x_v, $v = 1, \ldots, V$, are observed at regular time intervals of length Δ and corrupted by Gaussian noise. Thus, the observation matrix is of the form $\mathbf{M} = \mathbf{I}_V$ and the observations are given by

$$\mathbf{y}_n = \mathbf{x}_n + \mathbf{w}_n,\ n = 1, \ldots, N.$$

In the CO case the complete vector of observations $\mathbf{y} = [\mathbf{y}_1^\top, \ldots, \mathbf{y}_N^\top]^\top$ has dimension $VN \times 1$.

In the PO scenario only a linear combination of the proteins $x_P + 2x_{P_2}$ is observed at discrete time instants, also contaminated by Gaussian noise,

i.e., the observation matrix is given by $\mathbf{M} = [0, 1, 2, 0, 0]$ (with dimension $1 \times V$) and observations are generated as

$$y_n = x_{2,n} + 2x_{3,n} + w_n, \text{ where } w_n \sim \mathcal{N}_1(w_n; 0, \sigma^2).$$

In the PO case, a vector of scalar observations with dimension $N \times 1$ is constructed as $\mathbf{y} = [y_1, \ldots, y_N]^\top$.

To evaluate the performance of the PMCMC and the PNPMC methods we compute, in all the simulation runs, the MSE attained by the sample set that approximates the marginal posterior of $\boldsymbol{\theta}$, generated by both schemes. We compute the MSE of each parameter θ_k, $k = 1, \ldots, K$, based on the M-size final output as in equations (5.2) and (5.3) for the PMCMC and PNPMC algorithms, respectively. However, the MSE cannot be computed in real problems, where the true parameters θ_{k*} are unknown. To monitor the stability and the efficiency of the two sampling schemes based on the generated sample alone, we resort to the NESS computed as in equations (2.25) and (3.9), respectively.

6.4.3 Estimation of a unique parameter θ_1

In this section we present numerical results regarding the approximation of the posterior distribution $p(\theta_1, \mathbf{x}|\boldsymbol{\theta}_{\setminus 1}, \mathbf{y})$ of a unique rate parameter $\theta_1 = \log c_1$ and the populations $\mathbf{x}$, when the rest of parameters $\boldsymbol{\theta}_{\setminus 1} = [\theta_2, \ldots, \theta_K]^\top$, are assumed to be known.

We have performed $P = 100$ independent simulation runs of the PMCMC and the PNPMC schemes in the CO and the PO scenarios, with different (independent) population and observation vectors in each simulation. Both in the CO and the PO cases, the same true population trajectories $\mathbf{x}^{(p)}$, $p = 1, \ldots, P$, were used, but the observations in the CO scenario $\mathbf{y}_{CO}^{(p)}$ and in the PO scenario $\mathbf{y}_{PO}^{(p)}$ differ. The number of observation instants has been set to $N = 100$.

As a proposal pdf $q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(i-1)})$ for the PMCMC scheme we consider a Gaussian random walk update with variance $\sigma^2 = 1$. A total number of $I = 10^4$ iterations has been run in each simulation. A final sample of size $M = 10^3$ has been obtained from each Markov chain by discarding a burn-in period of 10^3 samples and thinning the output by a factor of 9. In the PNPMC scheme, the number of iterations has been set to $L = 10$, the number of samples per iteration is $M = 10^3$ and the clipping parameter is $M_T = 100$.

In Figure 6.4 the final MSE obtained by the PMCMC (*left*) and the PNPMC (*right*) algorithms for each simulation run is depicted versus the

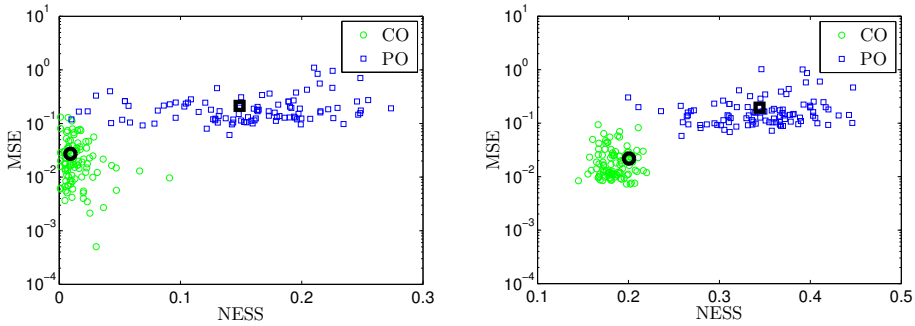

Figure 6.4: Final MSE versus final NESS obtained in each simulation run by the PMCMC (*left*) and the PNPMC (*right*) methods, in the CO and the PO scenario. The big markers represent average simulation runs.

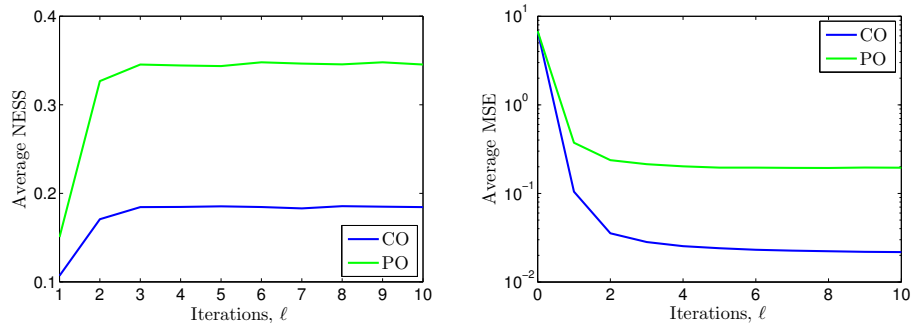

Figure 6.5: Evolution along the iterations of the PNPMC algorithm of the average NESS (*left*) and MSE (*right*) in the CO and PO scenarios.

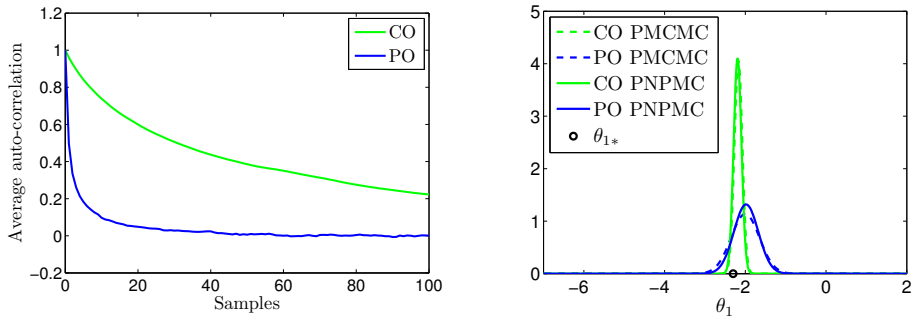

Figure 6.6: *Left*: Average ACF based on the final sample of size $M = 10^3$ of the PMCMC scheme in the CO and the PO scenarios. *Right*: Marginal posterior estimates $\hat{p}^{M,J}(\theta_1, |\boldsymbol{\theta}_{\setminus 1}, \mathbf{y})$ of an average simulation run, for PMCMC and PNPMC in the CO and PO scenarios.

final NESS, in the CO and the PO scenarios. Note that the NESS is computed differently for PMCMC and PNPMC. It can be observed that both algorithms perform similarly in this case, with an equivalent computational cost. Both algorithms attain on average lower MSE values in the CO scenario, as expected. However, the NESS also takes lower values in the CO case, which indicates a worse mixing of the Markov chains in the PMCMC algorithm and also higher degeneracy in the PNPMC algorithm.

In Figure 6.5 the evolution of the MSE (*right*) and the NESS (*left*) along the iterations of the PNPMC algorithm is represented, for the CO and the PO scenarios. It can be observed that both measures attain a steady value by the 5-th iteration, both in the CO and the PO case, which suggest that actually less iterations are sufficient for this problem. Again, we observe that in the CO scenario both the NESS and the MSE reach lower values.

Figure 6.6 (*left*) plots the average ACF of the final PMCMC sample, after removing the burn-in period and thinning the Markov chain by a factor of 9. Particularly high correlations are present in the CO case, leading to a poor NESS. Related to the ACF, the average sample acceptance probability in the PMCMC scheme in the PO scenario is 0.091, while in the CO scenario it is only 0.0034. Which means that 910 samples are accepted out of $I = 10^4$ in the CO case and only 34 in the CO case.

In Figure 6.6 (*right*) the final estimates $\hat{p}^{M,J}(\theta_1|\boldsymbol{\theta}_{\setminus 1}, \mathbf{y})$ of the average simulation runs represented as big circles and crosses in Figure 6.4 are represented in the CO and the PO scenario, for the PMCMC and the PNPMC schemes. For the PMCMC method we have built a Gaussian approximation of the posterior $p(\theta_1|\boldsymbol{\theta}_{\setminus 1}, \mathbf{y})$ based on the final MCMC sample $\{\theta_1^{(i)}\}_{i=1}^M$. For the PNPMC method, this approximation corresponds to the proposal pdf for the next iteration $L+1$, i.e., $\hat{p}^{M,J}(\theta_1|\boldsymbol{\theta}_{\setminus 1}, \mathbf{y}) = q_{L+1}(\theta_1) = \mathcal{N}(\theta_1; \mu_{L,1}, \sigma^2_{L,1})$, where the mean and variance terms $\mu_{L,1}$ and $\sigma^2_{L,1}$ are computed as in equation (3.2). It can be observed in Figure 6.6 that very similar results are obtained by both algorithms in this scenario. The final MSE values obtained by the PMCMC and the PNPMC methods, averaged over $P = 100$ simulation runs, are shown in Table 6.3, together with the MSE corresponding to the prior distribution.

Figure 6.7 depicts the posterior mean of the populations $\hat{\mathbf{x}}^{M,J} = E_{\hat{p}^{M,J}(\mathbf{x}|\mathbf{y})}[\mathbf{x}]$ obtained with PMCMC (*left*) and PNPMC (*right*) in the PO scenario. The results correspond to the particular simulation runs (different for PMCMC and PNPMC) represented with big squares in Figure 6.4 and whose posterior approximations $\hat{p}^{M,J}(\theta_1|\boldsymbol{\theta}_{\setminus 1}, \mathbf{y})$ are shown in Figure 6.6. It can be observed that, in the PO scenario, the tendency of the population of

Table 6.3: Final average MSE for θ_1 in the CO and PO scenarios, for PMCMC and PNPMC. The prior values are included for comparison.

	Prior	PO		CO	
		PMCMC	PNPMC	PMCMC	PNPMC
MSE θ_1	6.789	0.215	0.195	0.027	0.022

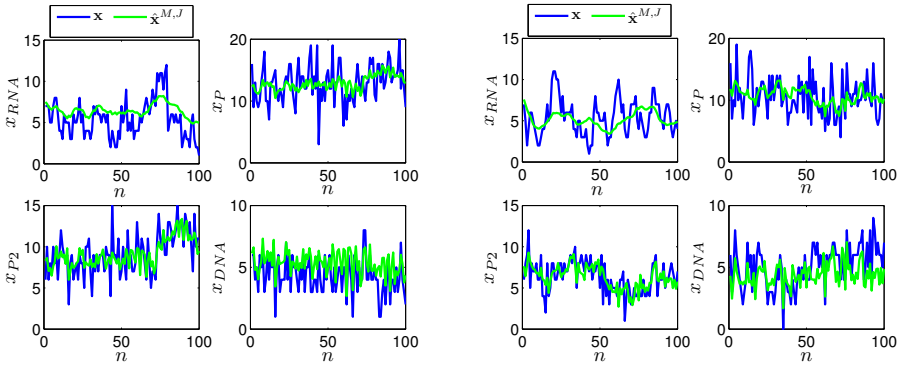

Figure 6.7: Posterior mean $\hat{\mathbf{x}}^{M,J} = E_{\hat{p}^{M,J}(\mathbf{x}|\mathbf{y})}[\mathbf{x}]$ of the populations obtained in a particular simulation run of PMCMC (*left*) and PNPMC (*right*) in the PO scenario.

all the species is reasonably identified, even though only a linear combination of the proteins is observed. In the CO scenario the populations of all species are precisely estimated and are not shown for conciseness. In these simulations we have performed exact sampling from the stochastic model with the Gillespie algorithm to obtain the likelihood approximation via a PF. Note that the populations of all species are very low, which suggests that the diffusion approximation may perform poorly in this scenario.

The results presented in this section reveal a very similar performance of the two analyzed methods in this simple scenario. Also in terms of computational complexity PMCMC and PNPMC perform very similarly. The execution time per 10^3 samples (one PNPMC iteration and 10^3 PMCMC iterations) for the PMCMC scheme is 312 seconds, while for PNPMC it is of 325 seconds, both in the CO and in the PO cases, on a 3-GHz Intel Core 2 Duo CPU, with 2 GB of RAM. The stochastic forward simulation of the prokaryotic model with the Gillespie algorithm has been implemented in C, and the rest of the code in Matlab R2007b.

However, the PMCMC method provides a set of highly correlated

samples, specially in the CO scenario, and requires the setting of the proposal variance σ^2 as well as the burn-in period length and the thinning parameter, which may not be straightforward and determines the performance of the algorithm. On the contrary, the PNPMC scheme provides uncorrelated sets of samples at each iteration, and does not require the precise fitting of any parameters (it is hardly sensitive to the choice of M_T). Additionally, the computer simulations suggest that the convergence of the PNPMC algorithm may be assessed observing the evolution of the NESS, which usually reaches a steady value simultaneously with the MSE.

6.4.4 Estimation of all the parameters θ_k, $k = 1, \ldots, K$

In this section we present simulation results to evaluate the performance of the PMCMC and the PNPMC schemes in the approximation of the posterior distribution of the rate parameters and the populations of all species, $p(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$, assuming that all the rate parameters are unknown, again in the CO and the PO scenarios.

In this case, $N = 200$ observation times are assumed for all the simulations. Again, $P = 100$ independent simulation runs of each algorithm have been performed. The PNPMC scheme has been run for $L = 15$ iterations, with $M = 10^3$ samples per iteration and clipping parameter $M_T = 100$. The PMCMC scheme has been run with $I = 15 \times 10^3$ iterations in each simulation run, a burn-in period of 10^3 iterations and thinning the output by a factor of 14.

In Figure 6.8 the MSE (in logarithmic scale), averaged over the parameters θ_k, attained by the PMCMC (*left*) and the PNPMC (*right*) algorithms is represented versus the NESS, in the CO and PO scenarios. Simulation runs which attained a final MSE close to the global average value are indicated with big circles (CO) and squares (PO) on both plots. It can be observed that the PMCMC method performs similarly in both scenarios, in terms of MSE and NESS, yielding poor results in both cases. On the contrary, the PNPMC method provides significantly better results in the CO scenario, where a larger amount of information is available. The PNPMC method does not present degradation due to the high degeneracy occurring in the CO scenario.

Figure 6.9 depicts the evolution along the iterations of the NESS (*left*) and the MSE (*right*) averaged over $P = 100$ independent simulation runs. Both measures converge to a steady value in a low number of iterations also in this complex scenario. As expected, a significantly higher final MSE is attained in the extremely data poor PO scenario.

In Figure 6.10 (*left*) the average ACF attained by the PMCMC in the CO and the PO cases is represented. Even after thinning the output, the sample correlation is extremely high in both scenarios, which leads to a very low NESS. The acceptance rate is also very low and very long chains are required to obtain reasonable results. In the PO scenario 43.69 samples are accepted on average in a simulation run of $I = 15 \times 10^3$ samples (acceptance rate 0.0029). In the CO case, only 23.07 samples are accepted on average (rate 0.0015).

Figure 6.10 (*right*) depicts the final Markov chain provided by the PMCMC method (after removing the burn-in period and thinning the output) in the average simulation run represented with a big square in Figure 6.8 (*left*). It can be observed that the mixing of the chain is very poor, with a total number of accepted samples of 46 (close to the average). Many other simulations, both in the PO and the CO scenarios, provide even lower numbers of accepted samples, and thus, very inconsistent results.

Figure 6.11 depicts the final Gaussian approximations of the marginal posteriors $p(\theta_k|\mathbf{y})$ obtained by the PMCMC and the PNPMC methods, in the CO and PO scenarios, for the average simulation runs represented as big circles and squares in Figure 6.8. We can observe that the PNPMC method provides a significantly better approximation of the log-rate parameters in the CO scenario, where a larger amount of data is available, which is also clear from Figure 6.8 (*right*). However, the PMCMC on average performs similarly in both scenarios, due to the low efficiency of the PMCMC sampling scheme when the dimension of the problem (either K or N) increases.

In Table 6.4 the MSE of each parameter θ_k averaged over $P = 100$ independent simulation runs is shown, obtained with PMCMC and PNPMC, for the CO and the PO experiments. In the CO case, PNPMC provides homogeneous results for all parameters. On the contrary, in the PO case, some of the parameters (specially θ_5 and θ_6) are significantly poorly estimated, presenting a final MSE close to the initial value (which corresponds to the prior knowledge). The PMCMC scheme presents significantly higher MSE values than PNPMC in both observation scenarios and for all parameters θ_k.

Figure 6.12 depicts the population posterior mean $\hat{\mathbf{x}}^{M,J} = E_{\hat{p}^{M,J}(\mathbf{x}|\mathbf{y})}[\mathbf{x}]$ corresponding to the average simulation runs of the PMCMC and the PNPMC methods in the PO scenario, represented as big squares in Figure 6.8. Again, the PNPMC method provides more accurate estimates of the unobserved populations than the PMCMC method, specially for x_{RNA}.

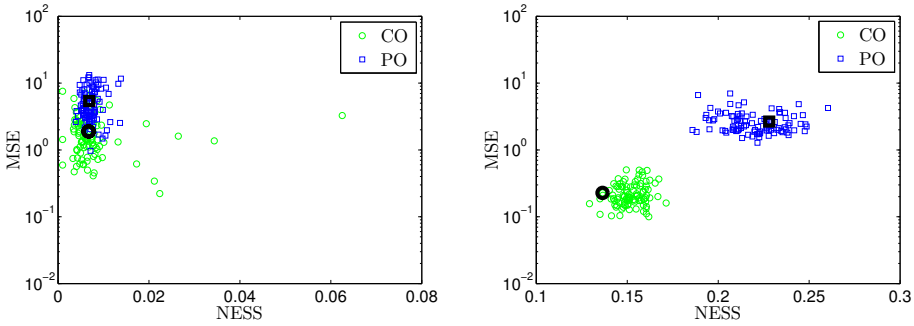

Figure 6.8: Final MSE versus final NESS, for each simulation run of the PMCMC (*left*) and the PNPMC (*right*) algorithms in the CO and the PO scenarios. The big markers represent average simulation runs.

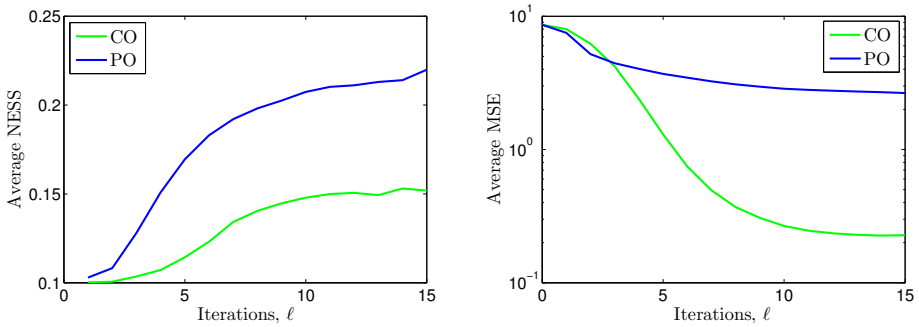

Figure 6.9: Evolution of the average NESS (*left*) and MSE (*right*) along the iterations of the PNPMC method in the CO and the PO scenario.

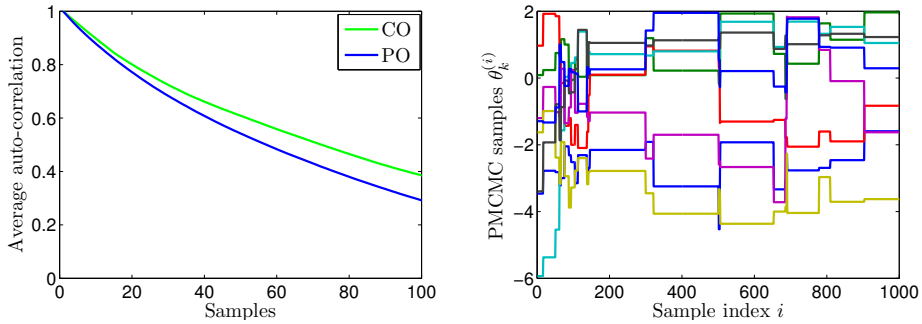

Figure 6.10: *Left*: Average ACF based on the final sample of size 10^3 of the PMCMC scheme in the CO and the PO scenarios. *Right*: Markov chain provided by the PMCMC method in the PO scenario, corresponding to the average simulation run depicted with a big square in Figure 6.8 (*left*).

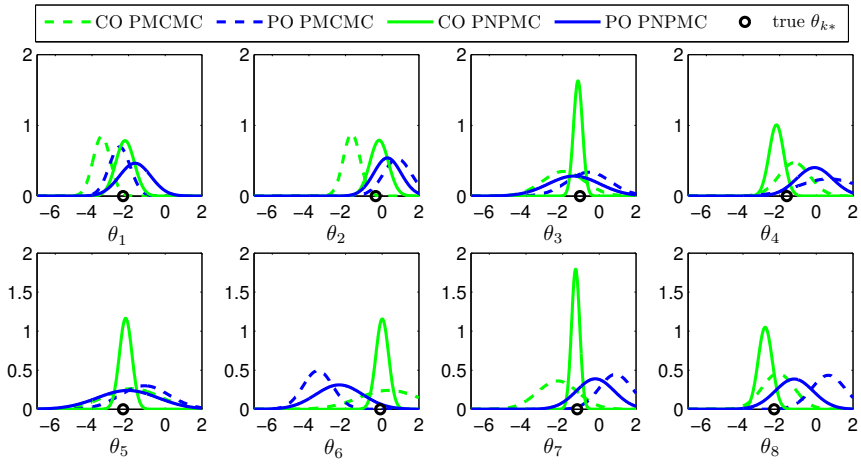

Figure 6.11: Marginal posterior approximations of each parameter $\hat{p}^{M,J}(\theta_k|\mathbf{y})$, $k = 1, \ldots, K$, attained in an average simulation run by the PMCMC and the PNPMC, in the CO and in the PO case.

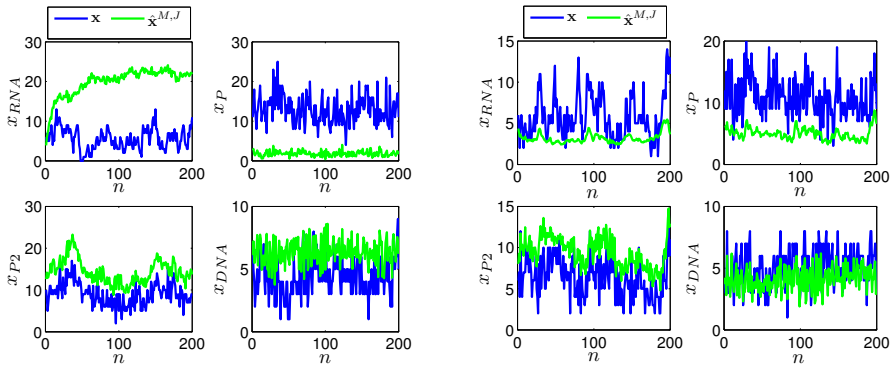

Figure 6.12: Posterior mean $\hat{\mathbf{x}}^{M,J} = E_{\hat{p}^{M,J}(\mathbf{x}|\mathbf{y})}[\mathbf{x}]$ of the populations of all species obtained in the average simulation run of the PMCMC (*left*) and the PNPMC (*right*) schemes, in the PO scenario.

Table 6.4: Final MSE for the parameters θ_k, $k = 1, \ldots, K$, in the CO and PO experiments, averaged over $P = 100$ simulation runs of the PMCMC and PNPMC algorithms.

	Prior	PO		CO	
		PMCMC	PNPMC	PMCMC	PNPMC
MSE θ_1	6.789	3.412	1.246	2.899	0.305
MSE θ_2	11.344	3.319	1.011	2.958	0.302
MSE θ_3	8.853	5.543	2.214	1.676	0.162
MSE θ_4	7.543	3.200	1.490	1.572	0.167
MSE θ_5	6.789	7.059	4.073	1.604	0.280
MSE θ_6	12.484	8.929	7.015	1.547	0.280
MSE θ_7	8.430	6.799	2.311	1.573	0.156
MSE θ_8	6.789	4.371	1.856	1.468	0.168
Average MSE	**8.628**	**5.329**	**2.652**	**1.912**	**0.228**

6.5 Conclusions

In this chapter we have applied the PNPMC method to the approximation of posterior distributions in SKMs. We provide computer simulations for a simple SKM known as predator-prey model, which allows to get insight into the complexity of the Bayesian inference problem in state-space models. We have compared the performance of the PNPMC method to the well known PMCMC method, applied to the challenging prokaryotic autoregulatory model. We have conducted computer experiments for the estimation of a single log-rate (Section 6.4.3) and all the log-rates (Section 6.4.4). The comparison for a single unknown rate allows to establish the performance of the methods in a "simple" problem and discard difficulties with the simulation code. The comparison with all log-rates unknown yields an assessment of the algorithm performance in more realistic (and challenging) scenarios. Additionally, we consider two observation scenarios, where a different amount of observations is available. We show that the complete observation case, where all of the species are observed, is computationally much more challenging, since the IWs present extreme variations, partly due to the PF approximation. The PNPMC algorithm, however, yields precise estimates of the parameters in this complex problem, opposite to the state of the art PMCMC algorithm.

Chapter 7

Bayesian inference in α-stable distributions

In this chapter we apply the NPMC algorithm to approximate the posterior probability distribution of the parameters of an α-stable random variable [138] given a set of independent realizations of the latter. This chapter is organized as follows. In Section 7.1 we provide an introduction to the basic concepts of α-stable distributions and we review the main existing techniques for the estimation of their parameters. The proposed NPMC inference algorithm is described in Section 7.2. Exhaustive computer simulations that illustrate the performance of the NPMC method as well as the main existing methods, based on synthetic data, are presented and discussed in Section 7.3. Numerical results obtained with a set of real fish displacement data are shown in Section 7.4. Finally, Section 7.5 is devoted to the conclusions of this chapter.

7.1 Introduction to α-stable distributions

The family of α-stable distributions [138] is a rich class of probability distributions that displays many patterns of shapes, allowing for asymmetry and heavy tails, opposite to the widely used, but more restrictive, Gaussian distribution. The class of α-stable distributions has been found suitable for statistical modelling in many different fields of sciences and engineering [134, 144, 111, 138]. For this reason, efficient computational algorithms for the estimation of the parameters of α-stable distributions in practical setups are needed.

A random variable is stable if a linear combination of two independent

copies of the variable has the same distribution, up to location and scale parameters. An α-stable distribution is a generalization of the Gaussian distribution and stems from a more general version of the central limit theorem, avoiding the assumption of finite variance [138].

We denote a general α-stable distribution as $\mathcal{S}(\alpha, \beta, \gamma, \delta)$, where $\alpha \in (0, 2]$ is a stability index (or characteristic exponent), $\beta \in [-1, 1]$ is a skewness parameter, and $\gamma > 0$ and $\delta \in \mathbb{R}$ determine the scale and location, respectively. The "shape" of the distribution is determined by α and β: lower values of α correspond to heavier tails and a sharper peak, while β determines the degree and sign of asymmetry ($\beta > 0$ corresponding to right-skewness). As $\alpha \to 2$, the distribution approaches a (non-standard) Gaussian distribution, and β becomes less meaningful and harder to estimate accurately. As $\alpha \to 0$, the effect of β becomes more pronounced, the density gets extremely high at the peak and the tails become heavier. Stable distributions have one single tail for $\alpha < 1$ and $\beta = \pm 1$, and both tails otherwise.

Distributions of the α-stable class have several specific mathematical properties. All (non-degenerate) stable distributions are unimodal, continuous and have an infinitely differentiable pdf [138]. However, the pdf is not available in closed form except for a few particular cases (Gaussian, Cauchy and Lévy) [138], a fact that has hampered a broader use of stable distributions in practice. The α-stable distribution is generally specified in terms of its characteristic function $\Phi(u) = E[\exp(iuX)]$, where X is the random variable and $i = \sqrt{-1}$. In this work we consider the so called 0-parameterization [138] of the characteristic function

$$\Phi(u) = \begin{cases} \exp\left\{i\delta u - \gamma^\alpha |u|^\alpha \left[1 + i\beta \tan(\frac{\pi\alpha}{2}) \text{sign}(u)(|\gamma u|^{(1-\alpha)} - 1)\right]\right\}, & \alpha \neq 1 \\ \exp\left\{i\delta u - \gamma|u| \left[1 + i\beta \frac{2}{\pi} \text{sign}(u) \log(\gamma|u|)\right]\right\}, & \alpha = 1 \end{cases} .$$

This parameterization is continuous in all the parameters and is more suitable for numerical work and statistical inference than alternative representations that can be found in the literature [138].

7.1.1 Simulation of univariate α-stable random variables

It is straightforward to generate samples from an α-stable distribution using an extension of the Box-Müller algorithm [138], which is detailed here for the 0-parameterization.

Let U and V be independent random variables, U uniformly distributed in the interval $(-\frac{\pi}{2}, \frac{\pi}{2})$ and V exponentially distributed with mean 1. For

any $0 < \alpha \leq 2$ and $-1 \leq \beta \leq 1$, when $\alpha \neq 1$, define $W = \frac{1}{\alpha} \arctan\left(\beta\left(\frac{\pi\alpha}{2}\right)\right)$. Then

$$Z = \begin{cases} \frac{\sin(\alpha(W+U))}{[\cos(\alpha W)\cos(U)]^{1/\alpha}} \left[\frac{\cos(\alpha W+(\alpha-1)U)}{V}\right]^{(1-\alpha)/\alpha}, & \text{if } \alpha \neq 1 \\ \frac{2}{\pi}\left[\left(\frac{\pi}{2}+\beta U\right)\tan(U) - \beta\log\left(\frac{\frac{\pi}{2}V\cos U}{\frac{\pi}{2}+\beta U}\right)\right], & \text{if } \alpha = 1 \end{cases}$$

has α-stable distribution $\mathcal{S}(z;\alpha,\beta,0,1)$ [138]. To simulate stable random variables $\mathcal{S}(x;\alpha,\beta,\gamma,\delta)$ with arbitrary scale and location parameters, the following transformation can be applied [138]

$$X = \begin{cases} \gamma[Z - \beta\tan(\frac{\pi\alpha}{2})] + \delta & \alpha \neq 1 \\ \gamma Z + \delta & \alpha = 1 \end{cases}.$$

7.1.2 Parameter estimation

A large number of methods for the estimation of the parameters of α-stable distributions have been proposed in the last decades, since the initial work of [60]. However, accurate estimation of all four parameters, specially when α is low, is still an open problem and an active area of research. The impossibility to evaluate the pdf associated to an α-stable distribution (except for a few particular cases), as well as the posterior dependencies among the parameters, makes the parameter estimation problem hard. In this work, we focus on the particular case in which only a small set of heavy-tailed observations are available, which further hinders the parameter estimation problem.

A computationally simple method based on data sample quantiles and look-up tables was proposed in [124], as a generalization of the method in [60], but it is known to yield consistent parameter estimates only when $0.4 \leq \alpha \leq 2$. In [138] a modified quantile method is proposed which is claimed to work for any values of the parameters. It allows to estimate low values of α, but yields poorer estimates of β than the standard quantile method. In [103], an iterative weighted regression procedure was introduced that fits the parameters to the empirical characteristic function (ECF) estimated from the data. This technique does not provide solutions for low values of α either. In [100] a simplified and improved version of the method in [103] is proposed which greatly reduces the amount of computation by restricting the estimation to an interval of the characteristic function. In [137] a maximum likelihood approach was proposed based on a numerical evaluation of the likelihood [135]. This method uses the quantile estimator of [124] as an initial approximation to the parameters

and maximizes the likelihood via an approximate gradient based search. It implements a fast likelihood evaluation but its use is restricted to cases when $\alpha > 0.4$. The fractional lower order moments and the log absolute moments methods have been proposed in [134] for the symmetric case ($\beta = \delta = 0$). Both methods are computationally simple but the latter has proved to be more efficient in practice [134]. Extensions of these methods have been proposed for the asymmetric case with $\delta = 0$ in [104]. These modified methods require transformations of the data into symmetrized and centered sequences, reducing the available sample size in one half and two thirds, respectively. When the amount of data is small, as considered here, this results in numerical difficulties and inconsistent estimates.

In the Bayesian framework, several attempts have been made to estimate the parameters of α-stable distributions by using MCMC algorithms. In [27] a Gibbs sampler is proposed, which requires sampling from a high-dimensional auxiliary variable and has, therefore, a high computational cost. The random walk MH sampler proposed in [118] relies on a likelihood approximation using the inverse fast Fourier transform (FFT) of the characteristic function near the mode and Bergström expansions for the tails. This sampler is very sensitive to the value of α, which determines the threshold between these two regions, as well as the spacing between the FFT samples. For this reason, it is very hard to tune the algorithm in such a way that acceptable results can be guaranteed for any α. In [140] a PRC-ABC method is proposed for Bayesian inference in univariate and multivariate α-stable models. This technique avoids the evaluation of the likelihood function using forward simulation from the α-stable distribution.

7.2 NPMC algorithm for Bayesian inference in α-stable models

Let $\boldsymbol{\theta} = [\alpha, \beta, \gamma, \delta]^\top$ be a vector containing the parameters of an α-stable distribution and let $\mathbf{y} = [y_1, \ldots, y_N]^\top$ be a vector of N i.i.d. samples from $\mathcal{S}(y; \alpha, \beta, \gamma, \delta)$, which denotes the pdf of the α-stable distribution. We adopt a Bayesian approach and aim at approximating the posterior probability distribution with density $p(\boldsymbol{\theta}|\mathbf{y})$ using a Monte Carlo scheme. The NPMC algorithm described in Section 3.3 can be readily applied to this problem (again, we only consider the clipping transformation of the IWs). However,

as the likelihood function

$$p(\mathbf{y}|\boldsymbol{\theta}) = \prod_{n=1}^{N} \mathcal{S}(y_n; \alpha, \beta, \gamma, \delta)$$

does not have a closed form and cannot be evaluated exactly, we resort to the numerical approximation proposed in [135]. Thus, the target distribution required to evaluate the standard IWs in step 3 of the NPMC algorithm (outlined in Table 3.2) is approximated as $\pi(\boldsymbol{\theta}_\ell^{(i)}) \approx \hat{p}(\mathbf{y}|\boldsymbol{\theta}_\ell^{(i)})p(\boldsymbol{\theta}_\ell^{(i)})$, where $\hat{p}(\mathbf{y}|\boldsymbol{\theta}_\ell^{(i)})$ denotes the likelihood approximation. The method in [135] provides an accurate approximation of general stable densities and distribution functions for essentially all values of the parameters. It is implemented in Nolan's STABLE program (available online) and in the Matlab function stblpdf, publicly available as part of the toolbox stbl in the web site www.mathworks.es. This toolbox uses an alternative parameterization of the characteristic function. Thus, a translation of the location parameter δ is needed in order to use this function under the 0-parameterization.

This approximation of the likelihood function introduces a distortion of the weights, additionally to the one due to the nonlinear transformation of the IWs. Assuming that the approximation error of the standard IWs is upper bounded by some constant $\epsilon \geq 0$, Theorem 1 in Section 4.3 yields an explicit error bound for the estimates produced by an importance sampler with approximate TIWs. The obtained upper bound consists of one term that vanishes as the number of samples $M \to \infty$ and another one proportional to the approximation error ϵ, which can only be reduced by increasing the computational effort of the approximation routine.

7.3 Computer simulations

In this section we provide extensive simulation results to illustrate the performance of the main existing methods for the estimation of α-stable parameters. The numerical results are obtained for a set of synthetic observations from α-stable distributions with a wide range of parameters. First we consider the NPMC and two other Bayesian methods: the MH and the PMC-ABC algorithms. The implemented NPMC and MH methods use the likelihood approximation proposed in [135], while the PMC-ABC method is based on a likelihood-free approach. Finally, we compare the NPMC method with the more relevant frequentist methods proposed in the literature.

7.3.1 Performance of the NPMC algorithm

We have performed 5000 independent simulations of the NPMC algorithm to approximate $p(\boldsymbol{\theta}|\mathbf{y})$ with different parameter and observation vectors. In each simulation run, we draw the parameters $\boldsymbol{\theta} = [\alpha, \beta, \gamma, \delta]^\top$ from a distribution $\mu(\boldsymbol{\theta}) = \mu(\alpha)\mu(\beta)\mu(\gamma)\mu(\delta)$ constructed from a set of independent uniform components, i.e.,

$$\begin{aligned} \mu(\alpha) &= \mathcal{U}(\alpha; (0, 2]), \\ \mu(\beta) &= \mathcal{U}(\beta; [-1, 1]), \\ \mu(\gamma) &= \mathcal{U}(\gamma; (0, 10]) \quad \text{and} \\ \mu(\delta) &= \mathcal{U}(\delta; [-5, 5]). \end{aligned}$$

Then, we generate a set of $N = 30$ samples y_n, $n = 1, \ldots, N$, from the resulting α-stable distribution $\mathcal{S}(\alpha, \beta, \gamma, \delta)$. We have selected such a low number of observations in order to reproduce as closely as possible the setup of the fish displacement dataset studied in Section 7.4, where around 30 observations are provided for each individual. We consider two different prior distributions for the inference algorithm. On the one hand, the actual prior distribution $p_1(\boldsymbol{\theta}) = \mu(\boldsymbol{\theta})$ defined above, and on the other hand, we consider a broader prior distribution for γ and δ, namely,

$$p_2(\gamma) = \mathcal{U}(\gamma; (0, 100]), \quad \text{and} \quad p_2(\delta) = \mathcal{U}(\delta; [-50, 50]),$$

to test the algorithm dependence on the choice of the prior distribution. We run the NPMC algorithm with $L = 10$ iterations in both settings, with $M = 300$ and $M_T = 20$ for $p_1(\boldsymbol{\theta})$, and $M = 10^3$ and $M_T = 30$ for $p_2(\boldsymbol{\theta})$.

At each iteration of the NPMC algorithm, $\ell = 1, \ldots, L$, we compute the $MSE_{\ell,k}$ associated to each parameter θ_k as in equation (5.3). The global MSE is obtained as $MSE_\ell = \frac{1}{4}\sum_{k=1}^{4} MSE_{\ell,k}$. We additionally compute at each iteration an approximation M_ℓ^{neff} of the NESS using equation (3.9), which serves as an indicator of the numerical stability of the algorithm.

In Figure 7.1, a smooth representation of the final MSE values (MSE_L) versus the final NESS (M_L^{neff}) values obtained at each of the 5000 simulation runs is shown. Results obtained with the narrow prior distribution $p_1(\boldsymbol{\theta})$ (*left*) and the broad prior distribution $p_2(\boldsymbol{\theta})$ (*right*) are displayed. A Gaussian kernel has been used to smooth the discrete sample representations. The big squares and circles represent simulation runs with a final MSE_L close to the global mean and median values, respectively. As it can be observed from the figure, in both cases the final NESS presents bimodality. A subset of the simulations ends up with a low NESS value,

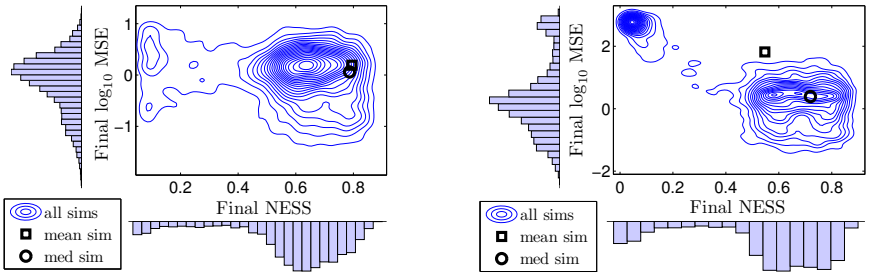

Figure 7.1: Smooth representation of the final MSE versus final NESS obtained by the NPMC algorithm in each simulation run, obtained with the p_1 (*left*) and p_2 (*right*) priors. Average and median simulation runs are depicted with big squares and circles, respectively.

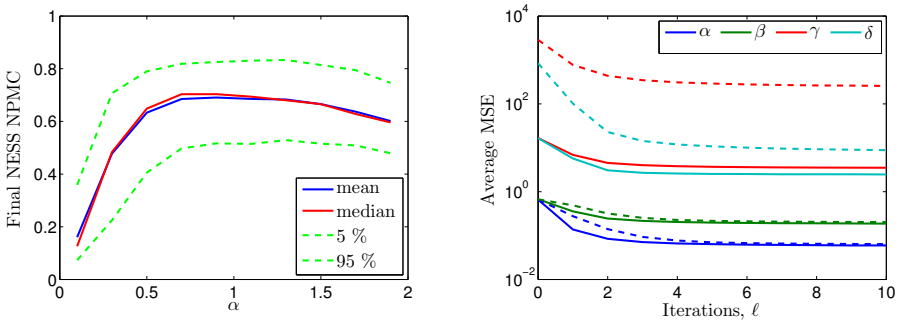

Figure 7.2: *Left*: Final NESS statistics versus the true value of α with the narrow prior distribution $p_1(\boldsymbol{\theta})$. *Right*: Evolution along the iterations of the MSE of each parameter, obtained with the narrow prior p_1 (solid lines) and broad prior p_2 (dashed lines).

yielding higher MSE values on average. When the broader prior is used, we obtain poorer performance, with a lower final NESS. However, when the final NESS is $M_L^{neff} > 0.3$, the performance is similar with both choices of the prior. These different behaviors are due to the value of parameter α, as will be made clear in the rest of this section.

In Figure 7.2 (*left*), some statistics (mean, median, 5% and 95% quantiles) of the final NESS value are represented versus the true value of α. The curves have been obtained from the final NESS values obtained at each simulation run for intervals of α of length 0.2. It can be observed that low α values (that is, stable distributions with heavy tails) yield low NESS values

after convergence of the algorithm. Very similar NESS results are obtained with the broader prior p_2. In Figure 7.2 (*right*) the evolution along the iterations of the MSE of each parameter ($MSE_{\ell,k}$) is represented, averaged over 5000 simulations runs, for the narrow (solid lines) and broad (dashed lines) prior distributions. The initial values $MSE_{0,k}$ have been obtained from the samples drawn from the prior $p(\boldsymbol{\theta})$ at the first iteration, before computing the IWs. It can be observed that the MSE smoothly decreases, reaching a steady value in a few iterations. Parameters α and β attain similar performance with both choices of the prior, since the corresponding marginal priors are the same under p_1 and p_2. However, parameters γ and δ attain a significantly poorer performance with the broader prior p_2. Specially, the γ parameter is estimated more poorly with the p_2 prior, on average.

7.3.2 Performance of the MH algorithm

In this section we consider a MH algorithm which, similarly to the NPMC method, uses the likelihood approximation proposed in [135]. Initially, we had implemented the MH method proposed in [118], which uses a likelihood approximation based on the inverse FFT of the characteristic function and Bergström expansions for the tails. However, this algorithm has turned out to be extremely sensitive to the selection of certain key parameters, such as the spacing between the FFT samples or the threshold between the two regions. It is fairly complicated, if not impossible, to adjust those parameters for a general case, particularly when the distribution of interest is heavy-tailed, as already noted in [140]. Provided that the method in [135] yields a good approximation to the likelihood for almost all values of the parameters (except for $\alpha \approx 0$), we have decided to run a standard MH algorithm which uses the likelihood approximation in [135] to compute the acceptance ratio.

In particular, we have applied the MH algorithm described in Table 2.6 with a Gaussian random walk proposal $q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(i-1)}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(i-1)}, \boldsymbol{\Sigma})$ with a covariance matrix $\boldsymbol{\Sigma} = \mathrm{diag}(0.25, 0.25, 1, 1)$. The acceptance probability reduces to

$$\min\left\{1, \frac{\hat{p}(\mathbf{y}|\boldsymbol{\theta}^\star)p(\boldsymbol{\theta}^\star)}{\hat{p}(\mathbf{y}|\boldsymbol{\theta}^{(i-1)})p(\boldsymbol{\theta}^{(i-1)})}\right\},$$

where $\hat{p}(\mathbf{y}|\boldsymbol{\theta}^\star)$ denotes the likelihood approximation computed with the method in [135].

The total chain length has been set to $I = 3000$ and $I = 10^4$ for the priors $p_1(\boldsymbol{\theta})$ and $p_2(\boldsymbol{\theta})$, respectively. This yields a total amount of processed samples equal to that of the NPMC method in Section 7.3.1. The bulk of the execution time of both techniques is the evaluation of the

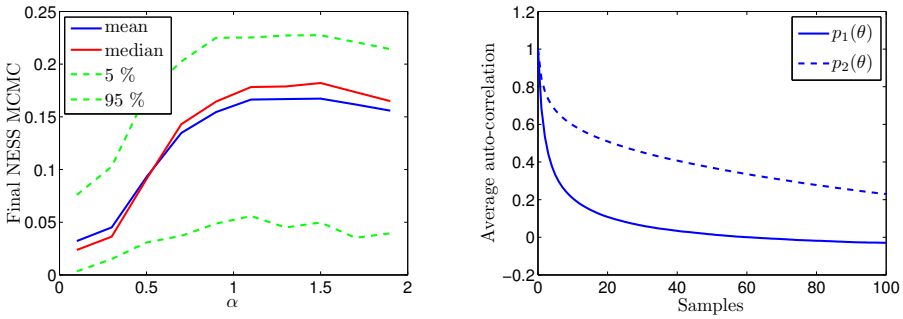

Figure 7.3: *Left*: NESS statistics versus the true value of α obtained by the MH algorithm with the prior distribution $p_1(\boldsymbol{\theta})$. *Right*: Average ACF of the final chains generated by the MH method, after removing the burn-in period and thinning the output, obtained with priors $p_1(\boldsymbol{\theta})$ and $p_2(\boldsymbol{\theta})$.

likelihood approximation for each sample $\boldsymbol{\theta}^{(i)}$, and thus both have a very similar computational complexity. The Markov chains generated by the MH algorithm have been post-processed, removing a burn-in batch of a 10 % of the number of samples I and then thinning by a factor of 9. Thus, we have obtained final sample sets of length $M = 300$ and $M = 10^4$ for the priors p_1 and p_2, respectively, the same as for the NPMC method. We have performed 5000 independent simulations with the same settings as for the NPMC algorithm in Section 7.3.1.

In Figure 7.3 (*left*), some statistics of the final average NESS obtained by the MH algorithm are represented versus the true value of α, for the prior distribution $p_1(\boldsymbol{\theta})$. Note that in the MCMC literature the ESS is defined differently from that used in IS techniques. In this case, it is an indicator of the size of a i.i.d. sample with the same variance as the current one, and is computed as in equation (2.25). It can be observed that, similarly to the results obtained with the NPMC method, for low values of α the performance of the algorithm is poorer. In this case, however, even for α values between 1 and 2, the NESS is around 15 %, which indicates that the resulting samples are highly correlated. Figure 7.3 (*right*) displays the average ACF obtained from the final Markov chains when either $p_1(\boldsymbol{\theta})$ or $p_2(\boldsymbol{\theta})$ are used as priors. It can be seen that the final samples obtained with the prior p_2 present a much higher correlation than with p_1 (the final sample size is also larger in this case).

7.3.3 Performance of the PMC-ABC algorithm

In [140] a PRC-ABC method is applied to the α-stable parameter estimation problem, which is claimed to outperform previous Bayesian attempts, such as the Gibbs sampler in [27] and the MH method in [118]. However, the PRC-ABC method described in [140] requires the setting of a large number of parameters, which affect the performance of the method and are very difficult to adjust for arbitrary α. We have performed simulations of the PRC-ABC method with the set of parameters suggested by the authors (with the summary statistics computed from the McCulloch's quantiles) and we have obtained highly inaccurate results for most values of α. The likelihood-free approximation is claimed to improve as the tolerance level ϵ decreases, but in practice it becomes inconsistent for low ϵ. As a stopping criterion, the authors propose to run 10 replicate sampler implementations, and to stop the algorithm when the NESS consistently drops below a given threshold, which results in a great increase in the computational complexity.

A comparison including the main ABC methods is provided in [155], which suggests that the PMC based scheme outlined in Table 2.12 is the one with the best performance. We have come to the same conclusions through our simulations and, for this reason, we include the PMC-ABC scheme in this comparison, instead of the PRC-ABC method of [140]. However, we have selected some of the parameters as suggested by [140].

At the first iteration, $\ell = 1$, the proposal distribution $q_1(\boldsymbol{\theta})$ is selected as the prior, and for iterations $\ell > 1$ it is constructed as $q_\ell(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\ell-1}, \boldsymbol{\Sigma}_{\ell-1})$, in the same manner as for the NPMC method. At each iteration $\ell = 1, \ldots, L$, pairs of samples $\boldsymbol{\theta}_\ell^{(i)} \sim q_\ell(\boldsymbol{\theta})$ and $\mathbf{y}_\ell^{(i)} \sim p(\mathbf{y}|\boldsymbol{\theta}_\ell^{(i)})$ are drawn until M samples are accepted. The metric ρ (the Euclidean distance in our case) is computed in terms of some nearly-sufficient low-dimensional summary statistics of the data, which we obtain as the quantile's method estimates of [124], as suggested in [140]. The scale parameter sequence has been set to $\epsilon_\ell \in \{100, 99, \ldots, 2, 1, 0.9, \ldots, 0.1\}$, with $L = 109$ iterations, and the number of samples per iteration has been set to $M = 1000$.

The acceptance rate becomes extremely low as the threshold parameter ϵ_ℓ decreases and, particularly, when α is low, which results in a high running time for the algorithm. For this reason, we have limited the execution of this method to 15 minutes (which is far more than the time required by the NPMC and the MH methods to converge under the same setting). We have performed 2500 independent simulation runs of this algorithm, with the prior distribution $p_1(\boldsymbol{\theta})$ only. Around 50 % of the simulations reached iteration $\ell = 100$.

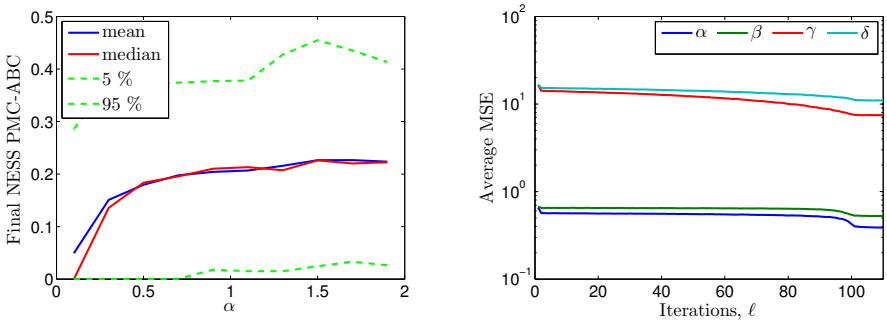

Figure 7.4: *Left*: Final NESS statistics versus the true α value obtained by the PMC-ABC algorithm with the prior pdf $p_1(\boldsymbol{\theta})$. *Right*: Average MSE along the iterations obtained by the PMC-ABC method with prior $p_1(\boldsymbol{\theta})$.

In Figure 7.4 we present the results obtained by the PMC-ABC method under the prior distribution $p_1(\boldsymbol{\theta})$. The *left* plot shows the statistics (mean, median, 5% and 95% quantiles) of the final NESS at the final iteration of the PMC-ABC algorithm. In this case, the NESS is computed in the same manner as for the NPMC method. It can be observed that, particularly for low α, the final NESS takes very low values, around 0.2 on average in the best case. In the *right* plot, the evolution of the average MSE is represented versus the iteration index ℓ. Only a slight improvement in terms of MSE can be observed along the iterations. If we further reduce the threshold ϵ_ℓ in order to improve the likelihood approximation, the computational time shoots up and the NESS values drop, leading to numerical instabilities. The results obtained with the broader prior $p_2(\boldsymbol{\theta})$ are extremely poor and have been omitted.

7.3.4 Comparison of the Bayesian methods

In Figure 7.5, the final average MSE of each parameter is represented versus the true value of α, as obtained by the NPMC and the MH methods with both prior choices $p_1(\boldsymbol{\theta})$ and $p_2(\boldsymbol{\theta})$, and by the PMC-ABC method only with prior $p_1(\boldsymbol{\theta})$. The MSE has been computed from the final sample, taking into account both the bias and the variance of the estimates, since the full posterior approximation allows to do so. It can be observed that both the NPMC and the MH techniques perform similarly with the prior distribution $p_1(\boldsymbol{\theta})$, except for $\alpha < 0.2$, where the NPMC attains better results. However, when the broader prior p_2 is considered, the MH algorithm yields highly inaccurate results due to the inefficiency of the Markov chains to explore the

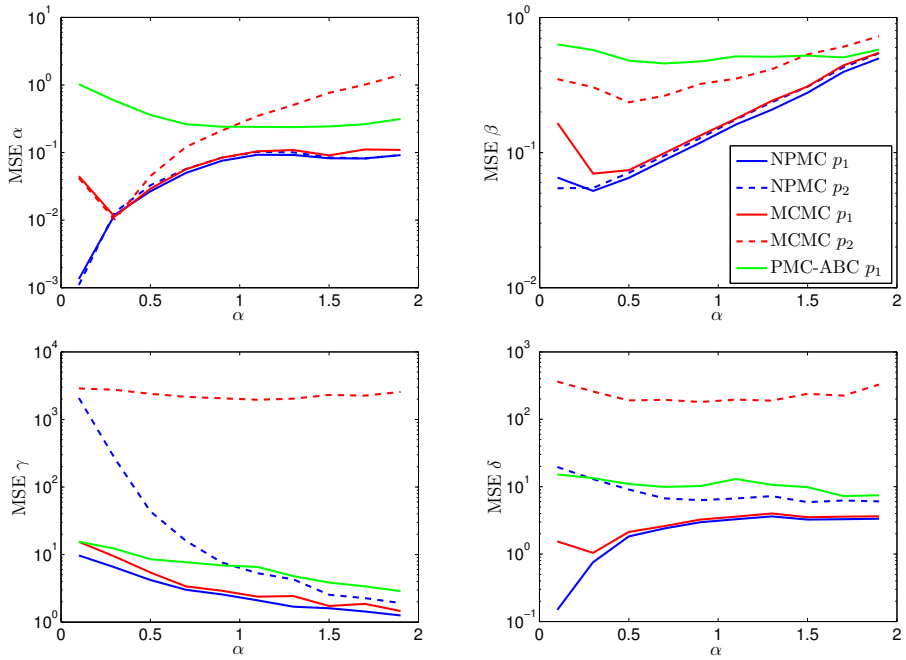

Figure 7.5: Average final MSE of each parameter versus the true value of α, obtained by the NPMC and MH methods, with the $p_1(\boldsymbol{\theta})$ and $p_2(\boldsymbol{\theta})$ prior distributions and 5000 independent simulations, and by the PMC-ABC method with $p_1(\boldsymbol{\theta})$ and 2500 simulations. The curves have been obtained by averaging the final MSE obtained in each simulation run in intervals of α of length 0.2.

broader space of $\boldsymbol{\theta}$ (which leads to low acceptance rates and high correlation among samples). This leads to a minor MSE reduction w.r.t. the prior distribution, specially for γ and δ. Much longer chains would be required to obtain reasonable results with this prior distribution. On the contrary, the NPMC method obtained similar MSE values in the estimation of α and β with both prior choices, for any value of α. The γ and δ parameters present significantly worse performance with the broader prior p_2, specially for low values of α. This reveals that with the low amount of observations considered in this setting ($N = 30$), the γ (and, to a lesser extent, δ) parameter cannot be identified when the distribution of interest presents very heavy tails. The selection of an informative prior for γ and δ leads to more efficient and robust algorithms, since it avoids the overestimation of these parameters, and allows to reduce the number of required samples. The likelihood-free

method performs poorly compared to any of the other Bayesian techniques, for all α. In view of these results, the NPMC algorithm appears to clearly outperform the other Bayesian methods.

7.3.5 Comparison with non-Bayesian methods

In this section we provide a comparison of the performance of the NPMC method with some of the main non-Bayesian methods proposed in the literature. Specifically, we consider the classical quantile method proposed in [124] (QT1), the modified quantile method in [138] (QT2), the ECF-based method of [100] (ECF), the ML estimation method of [137] (MLE) and the log-absolute moments method proposed in [134] (LAM). All of these techniques are implemented in the toolbox STABLE for different platforms, and provide point estimates $\hat{\theta}_k$ of the α-stable parameters from a set of observed data. We have performed 10^5 independent simulations of each of these methods and computed the empirical MSE from the point estimates as $MSE = \frac{1}{K}\sum_{k=1}^{K}(\hat{\theta}_k - \theta_k)^2$. For the NPMC method, the MSE has been computed in the same manner for this experiment, and thus the curves slightly differ from those shown in Figure 7.5. In the case of the NPMC technique we have considered 5000 simulation runs with the prior distribution $p_1(\boldsymbol{\theta})$. The simulation setup regarding the generation of the observed data fits the one described in Section 7.3.1.

In Figure 7.6, the average final MSE obtained by the various methods for each parameter is represented versus the true value of α. The LAM technique only provides estimates for α and γ. Regarding the estimation of the α parameter, the QT1, ECF and MLE methods are unable to estimate values of $\alpha < 0.4$. On the contrary, the NPMC, the QT2 and the LAM methods succeed to estimate low values of α. The NPMC method outperforms the other methods for all values of α, except for $\alpha \approx 2$, which corresponds to a Gaussian distribution. For the estimation of β the NPMC method also provides the best results, followed by the MLE and the QT1 methods. Given the low amount of observations, all of the methods fail to accurately estimate the true values of γ and δ for $\alpha < 0.5$, yielding the NPMC method the best (yet modest) results.

7.3.6 Remarks

The results presented in Sections 7.3.4 and 7.3.5 show that the NPMC method outperforms all the other methods we have studied (both Bayesian and frequentist) in terms of MSE for all α. Additionally, the NPMC method

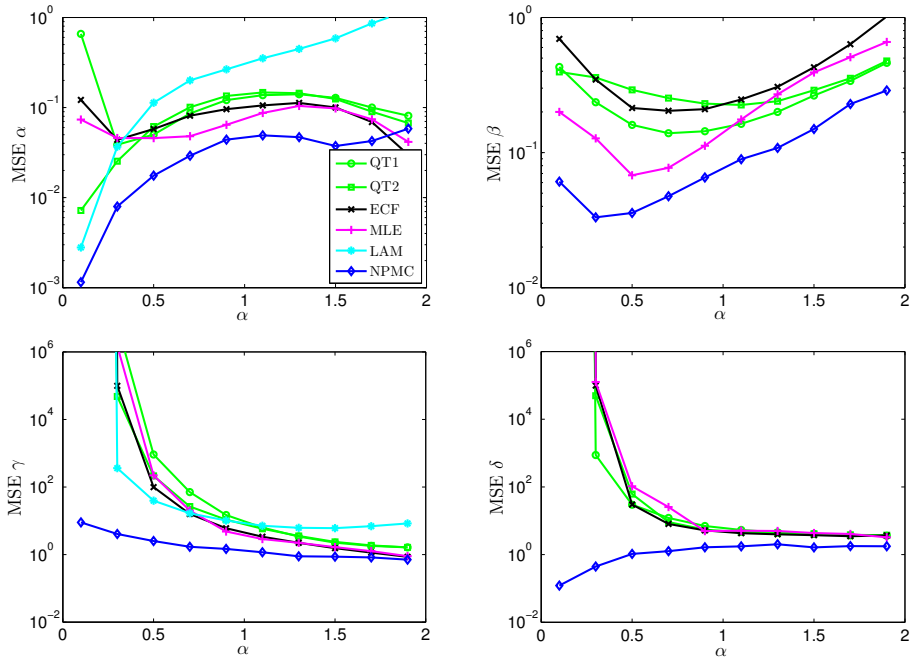

Figure 7.6: Average final MSE of each parameter versus the true value of α, obtained by the QT1, QT2, ECF, MLE, LAM and NPMC methods. The curves have been obtained by averaging the final MSE values obtained in each simulation run in intervals of α of length 0.2. The curves of the NPMC correspond to the narrow prior p_1.

is more robust to numerical issues, occurring mainly with low values of α. In Table 7.1 the failure rate[1] of each of the methods is shown, together with the corresponding execution times. The QT1, ECF, NPMC and MH methods are very robust to the α parameter value and only fail in around 0.35% of the simulations, when $\alpha < 0.01$. However, the MH algorithm performs poorly with the broader prior $p_2(\boldsymbol{\theta})$, yielding a high failure rate. The MLE method provides an error rate over 20% because the likelihood approximation routine implemented in STABLE does not work for $\alpha < 0.4$. The LAM method fails in 8% of the simulations, probably due to the low amount of observations considered, specially for heavy-tailed distributions. Finally, for the PMC-ABC method the failure rate is expressed in terms of the number of simulations that did not reach iteration $\ell = 50$ by the time

[1]The failure rate is defined as the percentage of simulation runs that end with a numerical error or warning indicating that the provided results are inaccurate.

Table 7.1: Failure rate and execution time of each algorithm.

	Failure rate (%)	Execution time
QT1	0.37	< 1 sec
QT2	5.05	< 1 sec
ECF	0.37	< 1 sec
MLE	21.1	< 1 sec
LAM	7.89	< 1 sec
NPMC	0.35	5 min
MH	0.5	5 min
PMC-ABC	27	15 min

limit of 15 minutes.

Regarding the execution times, Bayesian methods are significantly slower than the classical frequentist techniques. The NPMC and MH methods have similar computational complexity, while the ABC method is much slower. We have used the R version of STABLE 4.0 to run the non-Bayesian techniques included in the comparison. All Bayesian methods have been implemented in Matlab R2007b on a 3-GHz Intel Core 2 Duo CPU, with 2 GB of RAM. Contrary to the MH algorithm, in the case of NPMC and PMC-ABC, the processing of each sample in a given iteration can be easily parallelized to reduce the running time. Most of the execution time of the NPMC method is due to the likelihood approximation in [135]. Thus, the NPMC method is particularly efficient when the amount of observations is low, and in this case it provides a feasible alternative to standard frequentist methods. However, when a large number of observations is available, simpler methods may be sufficiently accurate, while Bayesian methods in general, and the NPMC method in particular, may result computationally too heavy.

It has to be noted that some of the frequentist methods, specially ECF and MLE, provide reasonable estimates of all 4 parameters with little computational complexity whenever $\alpha > 0.3$. However, the NPMC algorithm yields more accurate estimates in general, and performs significantly better for very low α, at the expense of an increase in the execution time.

In comparison with other Bayesian methods, the NPMC algorithm has clear advantages in terms of simplicity, estimation error and execution time. The NPMC method is straightforward to implement, and it only requires

a coarse selection of the parameters L, M and M_T. We propose to use $M_T \approx \sqrt{M}$, according to the theoretical analysis in Section 4.3. The convergence of the NPMC may be easily assessed in practice by observing the evolution of the NESS along the iterations, and stopping the adaptation when it reaches a steady value. Additionally, the NPMC method scales better as the complexity of the problem increases (a broader prior or a narrower likelihood, due to a larger number of observations).

7.4 Simulations with real fish displacement data

In this section we present the numerical results obtained with a set of real data describing fish displacement. Specifically, the data corresponds to the species *Salvelinus fontinalis* and was collected in Ganelon Creek, in Canada, in the summer of 1998. This data set was first described in [17].

7.4.1 Data description

The available set of observations $\mathbf{y}$ corresponds to the univariate daily displacement of $P = 21$ fish, measured in meters. The n-th displacement of the p-th fish, $y_{p,n}$, is defined as the position increment in one dimension between two consecutive measurements. The number of observations N_p associated to each individual is very low, taking values between 23 and 35. Figure 7.7 displays the available observations $y_{p,n}$ of three selected individuals, $p = 11, 20, 18$, at each time instant n. These three cases describe the typical behaviors present in the whole available data set.

Visual inspection of the available data reveals that it has no gaps in its support and presents unimodality, heavy-tails and asymmetry, and cannot be properly modeled by a Gaussian distribution [136]. Thus, we assume that these samples are independent and follow an α-stable distribution $y_{p,n} \sim \mathcal{S}(y; \alpha_p, \beta_p, \gamma_p, \delta_p)$. The individual $p = 11$ (*left* plot in Figure 7.7) presents a heavy-tailed and rather symmetric distribution, probably with a low value of α and β. Fish $p = 20$ (*central* plot) presents lighter tails and more asymmetry than the previous case. Finally, $p = 18$ (*right* plot) corresponds to a light-tailed and apparently symmetric distribution, similar to a Gaussian population.

7.4.2 Numerical results

We have applied the NPMC, the MH and the described frequentist methods to this problem and compared the obtained results. For the Bayesian

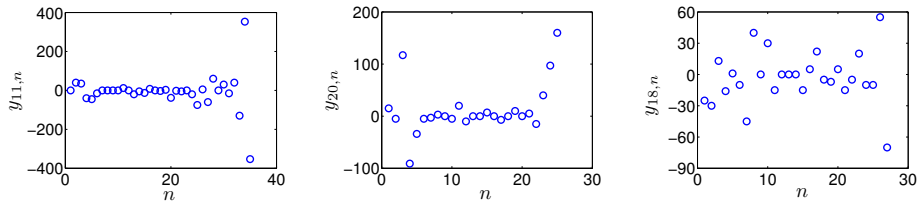

Figure 7.7: Real measurements of fish displacement $y_{p,n}$, $n = 1, \ldots, N_p$, of three selected individuals $p = 11$ (*left*), $p = 20$ (*central*) and $p = 18$ (*right*).

schemes, we have considered prior marginal distributions

$$p_3(\gamma) = \mathcal{U}(\gamma; (0, 50]) \quad \text{and} \quad p_3(\delta) = \mathcal{U}(\delta; [-10, 10]).$$

The parameters have again been set to $L = 10$, $M = 10^3$ and $M_T = 30$ for the NPMC method. In order to have a similar computational cost, the total number of iterations of the MH method has been set to $I = 10^4$, yielding a final sample of $M = 1000$ after removing the burn-in period and thinning.

Figure 7.8 shows the final NESS obtained by the NPMC (*left*) and the MH (*right*) methods, versus the corresponding values of α estimated by each algorithm, similarly to Figure 7.2 (*left*) and Figure 7.3 (*left*) in the computer simulations of Section 7.3. The particular cases $p = 11, 20, 18$, whose observations are shown in Figure 7.7, are depicted with big markers. It can be observed that similar results are obtained in the real data case, where low α values yield low final NESS. Since the NESS has proved to be a good indicator of the convergence of the NPMC method, and is related to the MSE evolution, it can be expected that in this real data problem the algorithm performs similarly to the example with synthetic data.

In Figure 7.9 the point estimates of the α, β, γ and δ parameters provided by the QT1, QT2, ECF, MLE, LAM, NPMC and MH methods are represented for the selected individuals $p = 11, 20, 18$. Additionally, a Gaussian approximation of the posterior distribution of each parameter is shown for the MLE (except for $p = 11$), NPMC and MH methods (obtained from the confidence intervals in the MLE case). As expected from the data inspection, the NPMC method identifies the case $p = 11$ as having a heavy-tailed distribution, with $\hat{\alpha}$ around 0.3, which is coherent with the LAM results, the other reliable method for estimating low α. The MLE method returns a final estimate of $\hat{\alpha} = 0.4$ and suggests via a warning message that the true value is actually lower. In the estimation of β, the NPMC and the MLE methods provide very similar results. The MH method yields similar α

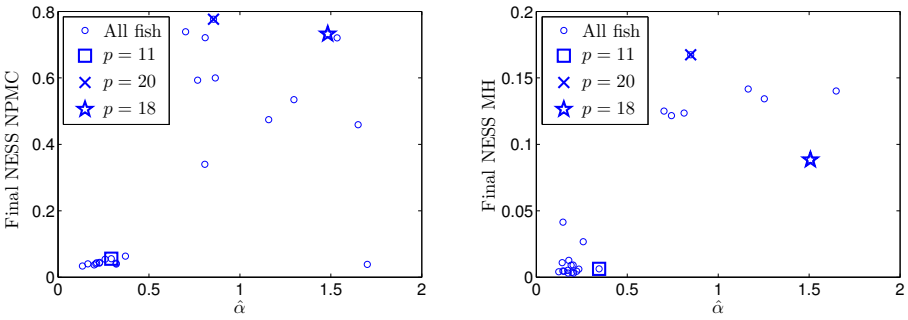

Figure 7.8: Final NESS obtained in each simulation by the NPMC (*left*) and the MH (*right*) algorithms, versus the corresponding estimates of α. Note that the NESS is computed differently in both cases. The vertical scale of the plots is also different.

and β estimates but with a larger variance. The estimate of γ is inaccurate in this case, but again the NPMC, LAM and MLE methods agree in their estimates. The NPMC and the MLE methods provide δ estimates close to 0. The MH method obtains very inaccurate estimates of γ and δ. In the case of $p = 20$, all methods agree to identify α as close to 0.8, except for the LAM method, which has shown to be less accurate when $\alpha > 0.5$ in the simulation study of Section 7.3. The estimates of the rest of parameters by the different methods are also similar. Finally, the last case $p = 18$ is identified as a light-tailed and symmetric distribution, close to a non-standard Gaussian. Table 3 summarizes the obtained results.

The consistency among the compared methods confirms that the available real data can be properly described by an α-stable distribution, as suggested by the visual data inspection. The numerical results are coherent with those obtained with synthetic data, both in terms of the NESS of NPMC and MH methods, and in terms of the comparison of the solutions provided by different techniques. The NPMC method provides consistent estimates (comparing different runs) of all parameters for all values of α, with an extremely low amount of observations. On the contrary, the MH algorithm fails to identify the parameters when α is low, as can be seen in Figure 7.5, and is very sensitive to the prior selection.

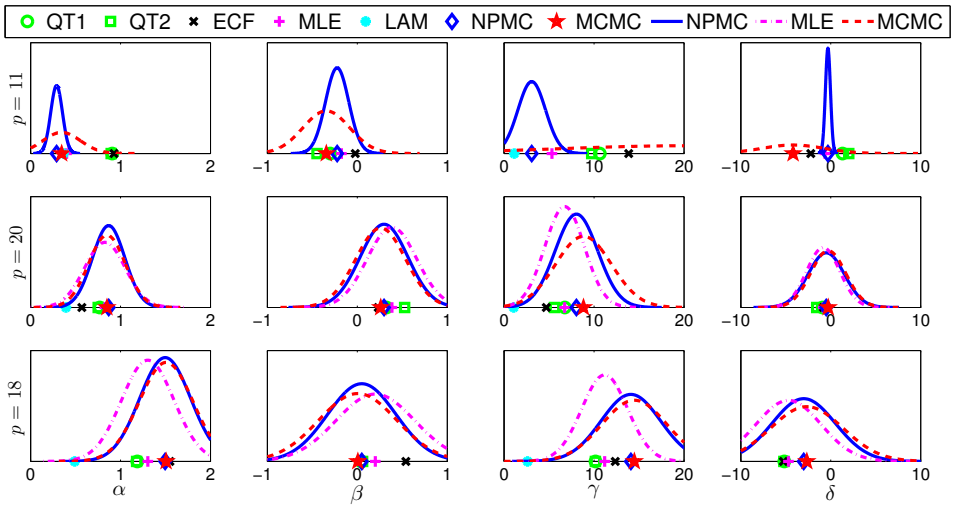

Figure 7.9: Point estimates of the α-stable parameters provided by the QT1, QT2, ECF, MLE, LAM, NPMC and MH methods, together with the Gaussian posterior approximation of the NPMC, MLE and MH methods, for $p = 11, 20, 18$. For $p = 11$ the MLE method does not yield confidence intervals, and thus the Gaussian posterior approximation is not shown.

Table 3: Point estimates of the parameters obtained by each of the methods, for the selected data sets $p = 11, 20, 18$. 95% confidence intervals are given in parentheses for MLE, NPMC and MH.

p	$\hat{\theta}_k$	QT1	QT2	ECF	MLE	LAM	NPMC	MH
11	$\hat{\alpha}$	0.91	0.89	0.93	0.40 (0)	0.37	0.29 (0.13)	0.34 (0.42)
	$\hat{\beta}$	-0.30	-0.45	-0.02	-0.18 (0)	0	-0.22 (0.25)	-0.34 (0.52)
	$\hat{\gamma}$	10.65	9.74	13.86	5.32 (0)	1.14	3.08 (3.01)	20.14 (28.56)
	$\hat{\delta}$	1.31	2.00	-2.22	-0.64 (0)	0	-0.31 (0.55)	-4.18 (6.94)
20	$\hat{\alpha}$	0.77	0.75	0.57	0.83 (0.44)	0.40	0.87 (0.35)	0.85 (0.41)
	$\hat{\beta}$	0.32	0.53	0.23	0.38 (0.54)	0	0.30 (0.52)	0.26 (0.56)
	$\hat{\gamma}$	6.76	5.61	4.70	6.74 (4.30)	1.08	8.04 (4.64)	8.83 (6.22)
	$\hat{\delta}$	-0.91	-1.56	-0.82	-0.84 (3.62)	0	-0.50 (3.97)	-0.29 (3.90)
18	$\hat{\alpha}$	1.19	1.18	1.55	1.30 (0.57)	0.49	1.50 (0.56)	1.51 (0.60)
	$\hat{\beta}$	0.04	0.06	0.54	0.20 (0.86)	0	0.05 (0.74)	0.01 (0.87)
	$\hat{\gamma}$	10.16	10.15	12.36	11.18 (5.01)	2.62	14.09 (6.47)	14.48 (7.23)
	$\hat{\delta}$	-5.16	-5.21	-5.26	-4.70 (7.10)	0	-3.01 (6.91)	-2.66 (8.08)

7.5 Conclusions

We have addressed the estimation of the parameters of α-stable distributions in a Bayesian framework. We have combined the proposed NPMC scheme with a classical numerical approximation of the α-stable pdf [135]. We have provided computer simulations with synthetic data comparing the NPMC method with the main techniques proposed in the literature for this problem. The NPMC algorithm clearly outperforms the traditional frequentist methods in terms of MSE, at the expense of a higher computation cost. It also yields better results than other Bayesian methods, such as MH or PMC-ABC methods, attaining a lower estimation error with a lower computational effort. Additionally, we have applied the studied methods to a fish displacement real dataset, and obtained coherent and satisfactory results. We have shown, by means of computer experiments, that the proposed technique attains a good performance even for small values of α and with an extremely low number of observations, where many of the existing techniques usually fail to perform adequately.

Chapter 8

Summary and future research lines

In this final chapter we summarize the contributions of this thesis and propose further extensions, in Sections 8.1 and 8.2 respectively.

8.1 Summary

The aim of this work has been the design, analysis and assessment of a novel family of Monte Carlo algorithms for the approximation of probability distributions. We have addressed the Bayesian inference problem and the approximation of posterior probability distributions by means of random samples. We have focused on the importance sampling (IS) approach, opposite to the widely used family of MCMC algorithms. The IS methodology has interesting features and important advantages over the MCMC approach. However, the main limitation of this algorithm is that it presents severe degeneracy of the importance weights (IWs) as the dimension of the model, K, and/or the number of observations, N, increase. This leads to a highly varying number of effective samples and inaccurate estimates, unless the number of samples is extremely high, which makes the method computationally prohibitive.

In this work we have investigated the population Monte Carlo (PMC) method, that consists in iteratively approximating a target distribution via an IS scheme. The same as standard IS, the PMC algorithm suffers from degeneracy of the IWs and, for this reason, the MCMC methodology is far more popular. The lack of a sufficient set of effective samples as a consequence of the weight degeneracy prevents from a robust proposal

update, yielding inaccurate estimates even in simple problems.

In this thesis we have introduced a new family of PMC algorithms that specifically addresses the degeneracy problem arising in IS techniques, and provides increased efficiency and robustness w.r.t. existing PMC and MCMC methods. The methodology revolves around a modification of the conventional IS principle that we term nonlinear IS (NIS), in reference to the nonlinear transformation applied to the classical IWs. We have analyzed the approximation of integrals using the NIS method and proved that they converge asymptotically, with explicit rates, in a number of settings. Additionally, we have conducted several computer simulation experiments that illustrate the proposed algorithms and we have compared the performance of the proposed scheme with that of powerful state of the art algorithms. In the next sections we summarize the proposed algorithms, the obtained theoretical results and the practical applications for which we have evaluated the performance of the proposed techniques.

8.1.1 NIS and NPMC with Gaussian proposals

The main contribution of this thesis is the introduction of the NIS technique, which, in addition to standard IWs, computes transformed IWs (TIWs) by means of a nonlinear transformation in order to reduce their fluctuations and thus avoid degeneracy. We have proposed two kinds of nonlinearities, of which the clipping transformation attains the best results in practice. We propose to apply this simple procedure in order to guarantee a prescribed ESS and a smooth and robust convergence.

The NIS scheme modifies the IWs and, hence, the standard theory of asymptotic convergence of IS (w.r.t. the number of samples) cannot be applied directly. To address this difficulty, we have analyzed the convergence of the approximations of integrals computed using clipped TIWs and proved that they converge a.s., similar to the results available for standard IS. We have also quantified the distortion introduced when using tempered TIWs. When the IWs cannot be computed exactly but they present a bounded approximation error, the convergence results provided in Section 4.3 apply.

It is straightforward to incorporate the new weight computation scheme into any existing method based on IS. Here we propose a nonlinear PMC (NPMC) algorithm which builds upon the advantages of the nonlinear transformation of the IWs. In Section 3.3 we have proposed a basic NPMC algorithm which constructs the proposal distributions as multivariate Gaussians. This choice of proposal distribution has been selected for simplicity and is not a restriction of the algorithm. This version of the

algorithm is better suited to problems where the target distribution can be reasonably described in terms of a Gaussian, that is, it is approximately unimodal and presents light tails.

In Section 5.1 we have numerically illustrated the principle behind the NIS and NPMC schemes and applied the proposed techniques to a simple simulation example, showing its great advantage over alternative methods. In the considered low-dimensional example, the degeneracy of the IWs is due to the large number of observations, which yields a very narrow likelihood function.

8.1.2 NPMC with mixture proposals

In Section 3.4 we have proposed an enhancement of the MPMC algorithm of [6], where the proposal densities are mixtures of Gaussian or Student's t kernels built by way of minimization of a KLD. We propose to update the importance function based on TIWs instead of standard IWs, in order to increase the efficiency of the underlying IS technique.

In Section 5.2 we have applied the proposed extension to solve a computational inference problem in a higher-dimensional space (where weight degeneracy is due to the curse of dimensionality). We have shown, through computer simulations, that the resulting nonlinear MPMC (NMPMC) algorithms drastically outperform their conventional MPMC counterparts, in terms of both estimation accuracy and robustness to numerical precision issues.

Additionally, we have proposed an extension that provides information about the number of components required to adequately represent the pdf of interest. In Section 5.3 we have compared the performance of the original and the proposed schemes in the cases of Gaussian and t mixtures, in two scenarios with a different number of samples (hence, with a different computational effort). We present numerical results that show that the proposed scheme clearly outperforms the original one.

8.1.3 Particle NPMC for state-space models

In Section 3.5 we have proposed a particle NPMC (PNPMC) method for the offline approximation of the joint posterior distribution of parameters and hidden states in state-space models. The proposed algorithm resorts to a particle filter (PF) approximation of the likelihood function to evaluate the importance weights, in an equivalent manner to the particle MCMC (PMCMC) algorithm. Additionally, it performs nonlinear transformations

of the weights to avoid degeneracy and the numerical problems typically arising in the proposal update of the PMC scheme in high-dimensional problems. In Section 4.4 we provide an extended convergence analysis of the NIS scheme, which takes into account the weight approximation. We have proved that, in this setting, integral approximation errors converge in L_2 with the usual Monte Carlo rate $\propto 1/\sqrt{M}$.

As a practical application, we have addressed the problem of approximating posterior distributions of the parameters and the populations in stochastic kinetic models. Initially, we have applied the proposed method to the approximation of the rate parameters in a predator-prey model. Additionally, we have compared the performance of the NPMC method to the well known PMCMC method, applied to the challenging prokaryotic autoregulatory model. We show how the NPMC scheme outperforms the PMCMC method with only a moderate computational cost.

8.1.4 NPMC for heavy-tailed distributions

Finally, we have addressed the estimation of the parameters of α-stable distributions in a Bayesian framework. We have combined the NPMC scheme of Section 3.3 with a classical numerical approximation of the α-stable pdf. The convergence results provided in Section 4.3 for NIS with approximate IWs apply in this case, yielding explicit and almost sure upper bounds for the approximation error.

In Section 7.3 we have provided computer simulations with synthetic data comparing the NPMC method with the main methods proposed in the literature for this problem. The NPMC algorithm clearly outperforms the traditional frequentist methods in terms of MSE, at the expense of a higher computation cost. It also yields better results than other Bayesian methods, such as the MH or the PMC-ABC algorithms, providing a smaller estimation error with a lower computational effort. Additionally, in Section 7.4 we have applied the studied methods to a fish displacement real data set, and obtained coherent and satisfactory results. We have shown, by means of computer experiments, that the proposed technique attains a good performance even for small values of α and with an extremely low number of observations, where many of the existing techniques usually fail to perform adequately.

8.1.5 Publications

The work contained in this thesis has led to the following publications in international journals and conferences: an initial version of the NPMC algorithm with clipping transformations of the likelihood function was proposed in [94]. The basic NPMC algorithm with Gaussian proposals and PF approximation of the likelihood function was introduced in [96]. The extensions of the MPMC algorithm with TIWs and adaptation of the number of mixture components were proposed in [95, 97]. In [98] we provide an extended convergence analysis and simulation results of the PNPMC algorithm for the prokaryotic model. In [99] we combine the NPMC algorithm with a numerical approximation of the α-stable pdf to estimate its parameters and provide a comparison with the main existing techniques. The convergence analysis of the NPMC algorithm with approximate IWs is also included in [99].

8.2 Future research lines

The work presented here can be extended and enhanced in different ways. Here we suggest some open research lines.

8.2.1 Convergence analysis of NPMC

In this thesis we have provided asymptotic convergence results for the NIS technique, which gives bounds for the approximation error of integrals as the number of samples M increases. An analysis of the convergence of the NPMC algorithm, which gives insight into the evolution of the approximation error along the iterations $\ell = 1, \ldots, L$ would be a most useful extension of the results reached so far. Based on numerical simulations, we expect that the NPMC presents a faster convergence to the target distribution than the standard PMC algorithm, both in the number of iterations and samples.

We are thus interested in upper bounds for the absolute approximation errors of integrals w.r.t. the random measure $\bar{\pi}_\ell^M$ constructed based on TIWs a each iteration $\ell = 1, \ldots, L$,

$$|(f, \bar{\pi}_\ell^M) - (f, \pi)| \leq R(M, M_T, \ell),$$

where $R(M, M_T, \ell)$ is a rate function that presumably converges toward 0 with M and ℓ.

8.2.2 NIS in filtering applications

The NIS technique can be applied in arbitrary IS-based methods, and the convergence results provided here apply as well. Thus, it is possible, for example, to use TIWs within the SMC samplers of [47], leading to a complete family of algorithms, of which the NPMC method introduced in the present paper would be just an instance. The NIS technique may also help to alleviate the degeneracy problems arising in very large scale and degenerate PF applications, where standard methods fail to perform adequately. However, the major benefits of this technique are reached in iterative IS settings, such as the PMC framework, where a large number of observations is available in a batch and the resulting likelihood function is extremely sharp. A preliminary work on this topic can be found in [127], where a PF with TIWs was proposed which aims at mitigating the degeneracy problem arising in sequential setups.

8.2.3 Efficient sampling in high dimensions

In the kind of problems addressed here the main limitation of the existing IS and PMC algorithms is the costly and time consuming evaluation of the likelihood function. The prior and proposal distributions are selected by the user and usually belong to standard families of well known distributions. For this reason, in moderately complex systems, the process of sampling and evaluating the proposal distribution is by no way the bottleneck in the computation process. However, we expect that in truly high-dimensional settings, the sampling and the evaluation of the proposal distribution can also become an issue. For this reason we propose to explore different proposal distributions, which allow for an efficient factorization, that enables an efficient sampling procedure.

For instance, in the NPMC algorithm of Section 3.3 the Gaussian proposal distribution can be decomposed into a number of univariate factors of the form

$$q_\ell(\boldsymbol{\theta}) = \prod_{k=1}^{K} q_\ell(\theta_k|\boldsymbol{\theta}_{\setminus k}).$$

This factorization enables the modeling of the proposal distribution at each iteration in terms of a Gaussian Bayesian network [132]. The main advantage of this graphical model for our purposes is the fact that it allows for a straightforward sampling procedure in spaces of arbitrarily high dimension. Indeed, a topological order of the variables of interest enables us to draw samples from them sequentially (one variable at a time) using the conditional

distribution of each variable given its ancestors. This approach is often termed ancestral sampling [9]. This idea has been proposed in [94] and is related to the Gibbs PMC algorithm in [50]. It would be interesting to explore this and other choices of efficient proposal distributions and evaluate their performance in real high-dimensional problems.

8.2.4 Parallel implementation for real applications

The simulation results provided in this work have been obtained with non-optimized Matlab code implemented in a regular personal computer. However, the proposed NPMC algorithms present a structure which is easy to implement in a distributed manner, opposite to other alternatives based on the MCMC principle. As already discussed, in the kind of applications addressed in this work, the main bulk of computational complexity of the proposed schemes is due to the evaluation or the approximation of the likelihood function for the computation of the IWs. This computations are straightforward to parallelize and perform in multiple cores or computers, which would result in a drastic decrease in execution time. In the NPMC algorithm in Table 3.2 only the two final steps require to be performed in a centralized manner, and they constitute a minor part of the computational complexity of the algorithm.

For this reason, we suggest to develop an optimized black-box parallel implementation of the NPMC algorithm, possibly using efficient programming languages as C or Python. This would allow to apply it to real, truly high-dimensional problems, such as those arising in different fields of science and engineering, for example, for autoregulatory networks, multiple target localization, ecology or meteorology, among others.

Appendix A

Acronyms and abbreviations

- ABC: approximate Bayesian computation
- AIS: annealed importance sampling
- CO: complete observation
- DPMC: D-kernel population Monte Carlo
- ECF: empirical characteristic function
- e.g.: exempli gratia (for instance)
- ESS: effective sample size
- GMM: Gaussian mixture model
- IBIS: iterative batch importance sampling
- i.e.: id est (that is)
- i.i.d.: independent and identically distributed
- IS: importance sampling
- IW: importance weight
- KLD: Kullback-Leibler divergence
- LAM: log absolute moments
- MAP: maximum *a posteriori*

- MCMC: Markov chain Monte Carlo
- MH: Metropolis-Hastings
- ML: maximum likelihood
- MLE: maximum likelihood estimator
- MMSE: minimum mean-square error
- MPMC: mixture population Monte Carlo
- MSE: mean square error
- MultiPMC: multiple population Monte Carlo
- NESS: normalized effective sample size
- NPMC: nonlinear population Monte Carlo
- NIS: nonlinear importance sampling
- PF: particle filter
- PMC: population Monte Carlo
- PMCMC: particle Markov chain Monte Carlo
- PNPMC: particle nonlinear population Monte Carlo
- pdf: probability density function
- PO: partial observation
- QT: quantile
- PRC: partial rejection control
- RB: Rao-Blackwellization
- SIS: sequential importance sampling
- SKM: stochastic kinetic model
- SMC: sequential Monte Carlo
- TIW: transformed importance weight
- w.r.t.: with respect to

Appendix B

Notation

- $\boldsymbol{\theta} = [\theta_1, ..., \theta_K]^\top \in \mathbb{R}^K$: parameter vector random variable
- $\boldsymbol{\theta}_*$: true parameter vector
- $\mathbf{x} = [\mathbf{x}_0^\top, \ldots, \mathbf{x}_N^\top]^\top$: hidden state with $N+1$ components $\mathbf{x}_n \in \mathbb{R}^V$
- $\mathbf{y} = [\mathbf{y}_1^\top, \ldots, \mathbf{y}_N^\top]^\top$: observation vector with N components $\mathbf{y}_n \in \mathbb{R}^D$
- $p(\boldsymbol{\theta})$: prior pdf of $\boldsymbol{\theta}$
- $\pi(\boldsymbol{\theta})$: target pdf of $\boldsymbol{\theta}$
- $q(\boldsymbol{\theta})$: proposal pdf or importance function
- $p(\mathbf{y}|\boldsymbol{\theta})$: conditional pdf of $\mathbf{y}$ given $\boldsymbol{\theta}$ (likelihood function)
- $\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$: a random variable or a sample $\boldsymbol{\theta}$ has a distribution $p(\boldsymbol{\theta})$
- $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^M$: set of M samples of $\boldsymbol{\theta}$
- $\delta_{\boldsymbol{\theta}^{(i)}}(d\boldsymbol{\theta})$: unit delta measure located at $\boldsymbol{\theta}^{(i)}$
- $\boldsymbol{\theta}^\star$: proposed sample in MCMC algorithm
- $w^{(i)}$, $w^{(i)*}$: standard normalized and unnormalized IW of a sample $\boldsymbol{\theta}^{(i)}$
- $\bar{w}^{(i)}$, $\bar{w}^{(i)*}$: normalized and unnormalized TIW of a sample $\boldsymbol{\theta}^{(i)}$
- $\omega_n^{(j)}$, $\omega_n^{(j)*}$: normalized and unnormalized IWs of a particle $\mathbf{x}_n^{(j)}$ in PF
- L, I: number of iterations in PMC and MCMC, respectively

- φ^M: nonlinear transformation function of the IWs
- M_T: clipping parameter
- $\mathcal{U}(\boldsymbol{\theta}; [a, b])$: uniform pdf of $\boldsymbol{\theta}$ in the interval between a and b
- $\mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$: Guassian pdf of $\boldsymbol{\theta}$, with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
- $E_{\pi(\boldsymbol{\theta})}[f(\boldsymbol{\theta})]$: expectation of a function $f(\boldsymbol{\theta})$ w.r.t. the pdf $\pi(\boldsymbol{\theta})$
- (f, π): integral of function f w.r.t. the pdf π
- $\pi^M(d\boldsymbol{\theta})$: M-sample discrete random measure approximating $\pi(\boldsymbol{\theta})$
- M^{eff}, M^{neff}: ESS and NESS

Bibliography

[1] K. Achutegui, J. Míguez, J. Rodas, and C. J. Escudero. A multi-model sequential Monte Carlo methodology for indoor tracking: Algorithms and experimental results. *Signal Processing*, 92(11):2594–2613, 2012.

[2] B. D. O. Anderson and J. B. Moore. *Optimal filtering*. Courier Corporation, 2012.

[3] J. L. Anderson and S. L. Anderson. A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, 127(12):2741–2758, 1999.

[4] C. Andrieu, N. De Freitas, and A. Doucet. Sequential MCMC for Bayesian model selection. In *Higher-Order Statistics, 1999. Proceedings of the IEEE Signal Processing Workshop on*, pages 130–134. IEEE, 1999.

[5] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

[6] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

[7] C. Andrieu, A. Doucet, S. S. Singh, and V. B. Tadic. Particle methods for change detection, system identification, and control. *Proceedings of the IEEE*, 92(3):423–438, 2004.

[8] C. Andrieu, A. Doucet, and V. B. Tadic. On-line parameter estimation in general state-space models. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 332–337. IEEE, 2005.

[9] D. Angelova and L. Mihaylova. Extended object tracking using Monte Carlo methods. *Signal Processing, IEEE Transactions on*, 56(2):825–832, 2008.

[10] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002.

[11] A. Baggio and K. Langendoen. Monte Carlo localization for mobile wireless sensor networks. *Ad Hoc Networks*, 6(5):718–733, 2008.

[12] A. Bain and D. Crisan. *Fundamentals of stochastic filtering*, volume 60. Springer Verlag, 2008.

[13] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[14] M. A. Beaumont. Approximate Bayesian computation in evolution and ecology. *Annual review of ecology, evolution, and systematics*, 41:379–406, 2010.

[15] M. A. Beaumont, J. M. Cornuet, J. M. Marin, and C. P. Robert. Adaptive approximate Bayesian computation. *Biometrika*, 96(4):983–990, 2009.

[16] M. A. Beaumont, W. Zhang, and D. J. Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

[17] G. Bélanger and M. A. Rodríguez. Homing behaviour of stream-dwelling brook charr following experimental displacement. *Journal of Fish Biology*, 59(4):987–1001, 2001.

[18] R. Bellazzi, P. Magni, and G. De Nicolao. Bayesian analysis of blood glucose time series from diabetes home monitoring. *Biomedical Engineering, IEEE Transactions on*, 47(7):971–975, 2000.

[19] T. Bengtsson, P. Bickel, and B. Li. Curse of dimensionality revisited: Collapse of particle filter in very large scale systems. *Probability and statistics: Essay in honour of David A. Freedman*, 2:316–334, 2008.

[20] A. Beskos, D. Crisan, and A. Jasra. On the stability of sequential Monte Carlo methods in high dimensions. *The Annals of Applied Probability*, 24(4):1396–1445, 2014.

[21] P. Bickel, B. Li, and T. Bengtsson. Sharp failure rates for the bootstrap particle filter in high dimensions. In *Pushing the limits of contemporary statistics: Contributions in honor of Jayanta K. Ghosh*, pages 318–329. Institute of Mathematical Statistics, 2008.

[22] K. Binder and D. Heermann. *Monte Carlo simulation in statistical physics: an introduction*. Springer Science & Business Media, 2010.

[23] C. M. Bishop. *Pattern recognition and machine learning*, volume 4. Springer New York, 2006.

[24] P. Boyle, M. Broadie, and P. Glasserman. Monte Carlo methods for security pricing. *Journal of economic dynamics and control*, 21(8):1267–1321, 1997.

[25] P. P. Boyle. Options: A Monte Carlo approach. *Journal of financial economics*, 4(3):323–338, 1977.

[26] R. J. Boys, D. J. Wilkinson, and T. B. L. Kirkwood. Bayesian inference for a discretely observed stochastic kinetic model. *Statistics and Computing*, 18(2):125–135, 2008.

[27] D. J. Buckle. Bayesian inference for stable distributions. *Journal of the American Statistical Association*, 90(430):605–613, 1995.

[28] M. F. Bugallo, M. Hong, and P. M. Djuric. Marginalized population Monte Carlo. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 2925–2928. IEEE, 2009.

[29] D. E. Burmaster and P. D. Anderson. Principles of good practice for the use of Monte Carlo techniques in human health and ecological risk assessments. *Risk analysis*, 14(4):477–481, 1994.

[30] O. Cappé, R. Douc, A. Guillin, J. M. Marin, and C. P. Robert. Adaptive importance sampling in general mixture classes. *Statistics and Computing*, 18(4):447–459, 2008.

[31] O. Cappé, S.J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

[32] O. Cappé, A. Guillin, J. M. Marin, and C. P. Robert. Population Monte Carlo. *Computational and Graphical Statistics*, 13(4):907–929, 2004.

[33] J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for nonlinear problems. In *Radar, Sonar and Navigation, IEE Proceedings-*, volume 146, pages 2–7. IET, 1999.

[34] G. Casella and E. I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

[35] G. Casella and C. P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

[36] G. Celeux, J. M. Marin, and C. P. Robert. Iterated importance sampling in missing data problems. *Computational statistics & data analysis*, 50(12):3386–3404, 2006.

[37] F. Cérou, P. Del Moral, and A. Guyader. A nonasymptotic theorem for unnormalized Feynman-Kac particle models. *Ann. Inst. Henri Poincaré Probab. Stat*, 47(3):629–649, 2011.

[38] M. H. Chen, J. G. Ibrahim, and Q. M. Shao. *Monte Carlo methods in Bayesian computation*. Springer, 2000.

[39] R. Chen and J. S. Liu. Mixture kalman filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(3):493–508, 2000.

[40] S. Chib, F. Nardari, and N. Shephard. Markov chain Monte Carlo methods for stochastic volatility models. *Journal of Econometrics*, 108(2):281–316, 2002.

[41] N. Chopin. A sequential particle filter method for static models. *Biometrika*, 89(3):539–552, 2002.

[42] N. Chopin, P. E. Jacob, and O. Papaspiliopoulos. SMC^2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):397–426, 2013.

[43] J. Cornuet, J. M. Marin, A. Mira, and C. P. Robert. Adaptive multiple importance sampling. *Scandinavian Journal of Statistics*, 39(4):798–812, 2012.

[44] D. Crisan and J. Míguez. Nested particle filters for online parameter estimation in discrete-time state-space Markov models. *arXiv preprint arXiv:1308.1883v2*, 2014.

[45] D. Crisan and J. Míguez. Particle-kernel estimation of the filter density in state-space models. *Bernoulli*, 20(4):1879–1929, 2014.

[46] M. H. DeGroot and M. J. Schervish. Probability and statistics. 2002.

[47] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[48] P. Del Moral, A. Doucet, and A. Jasra. An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing*, 22(5):1009–1020, 2012.

[49] O. Demirel, I. Smal, W. J. Niessen, E. Meijering, and I. F. Sbalzarini. Piecewise constant sequential importance sampling for fast particle filtering. In *Data Fusion & Target Tracking 2014: Algorithms and Applications (DF&TT 2014), IET Conference on*, pages 1–8. IET, 2014.

[50] P. M Djuric, Shen B., and M. F. Bugallo. Population Monte Carlo methodology a la Gibbs sampling. In *EUSIPCO*, 2011.

[51] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez. Particle filtering. *IEEE Signal Processing Magazine*, 20(5):19–38, 2003.

[52] R. Douc, A. Guillin, J. M. Marin, and C. P. Robert. Convergence of adaptive mixtures of importance sampling schemes. *The Annals of Statistics*, 35(1):420–448, 2007.

[53] R. Douc, A. Guillin, J. M. Marin, and C. P. Robert. Minimum variance importance sampling via population Monte Carlo. *ESAIM: Probability and Statistics*, 11:427–447, 2007.

[54] A. Doucet, N. De Freitas, and N. Gordon. *An introduction to sequential Monte Carlo methods*. Springer, 2001.

[55] A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer Verlag, 2001.

[56] A. Doucet, N. De Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 176–183. Morgan Kaufmann Publishers Inc., 2000.

[57] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[58] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222, 1987.

[59] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans (1978–2012)*, 99(C5):10143–10162, 1994.

[60] E. F. Fama and R. Roll. Parameter estimates for symmetric stable distributions. *Journal of the American Statistical Association*, 66(334):331–338, 1971.

[61] P. Fearnhead. Computational methods for complex stochastic systems: a review of some alternatives to MCMC. *Statistics and Computing*, 18(2):151–171, 2008.

[62] G. S. Fishman. *Monte Carlo*. Springer-Verlag New York, 1996.

[63] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*, volume 2. Taylor & Francis, 2014.

[64] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):721–741, 1984.

[65] J. Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica: Journal of the Econometric Society*, pages 1317–1339, 1989.

[66] W. R. Gilks. *Markov chain Monte Carlo*. Wiley Online Library, 2005.

[67] W. R. Gilks and C. Berzuini. Following a moving target - Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146, 2001.

[68] W. R. Gilks, G. O. Roberts, and E. I. George. Adaptive direction sampling. *The statistician*, pages 179–189, 1994.

[69] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[70] P. Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer, 2004.

[71] A. Golightly and D. J. Wilkinson. Bayesian inference for stochastic kinetic models using a diffusion approximation. *Biometrics*, 61(3):781–788, 2005.

[72] A. Golightly and D. J. Wilkinson. Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. *Interface Focus*, 1(6):807–820, 2011.

[73] N. J. Gordon, D. J. Salmond, and A. FM Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.

[74] R. Gramacy, R. Samworth, and R. King. Importance tempering. *Statistics and Computing*, 20(1):1–7, 2010.

[75] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[76] R. C. Griffiths and S. Tavare. Monte Carlo inference methods in population genetics. *Mathematical and computer modelling*, 23(8):141–158, 1996.

[77] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[78] M. Hong, M. F. Bugallo, and P. M. Djuric. Joint model selection and parameter estimation by population Monte Carlo simulation. *Selected Topics in Signal Processing, IEEE Journal of*, 4(3):526–539, 2010.

[79] D. W. Hubbard. *The failure of risk management: Why it's broken and how to fix it*. John Wiley and Sons, 2009.

[80] A. Iacobucci, J. M. Marin, and C. P. Robert. On variance stabilisation in population Monte Carlo by double Rao-Blackwellisation. *Computational Statistics & Data Analysis*, 54(3):698–710, 2010.

[81] Y. Iba. Population-based Monte Carlo algorithms. *Trans. Jpn. Soc. Artif. Intell.*, 16(cond-mat/0008226. ISM-757. 2):279–286. 6 p, Aug 2000.

[82] P. Jäckel and R. Bubley. *Monte Carlo methods in finance*. J. Wiley, 2002.

[83] A. Jasra and P. Del Moral. Sequential Monte Carlo methods for option pricing. *Stochastic analysis and applications*, 29(2):292–316, 2011.

[84] A. Jasra, D. A. Stephens, A. Doucet, and T. Tsagaris. Inference for Lévy-driven stochastic volatility models via adaptive sequential Monte Carlo. *Scandinavian Journal of Statistics*, 38:1–22, 2011.

[85] A. Jasra, D. A. Stephens, and C. C. Holmes. On population-based simulation for static inference. *Statistics and Computing*, 17(3):263–279, 2007.

[86] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[87] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

[88] N. Kantas, A. Doucet, S. S. Singh, and J. M. Maciejowski. An overview of sequential Monte Carlo methods for parameter estimation in general state-space models. In *15th IFAC Symposium on System Identification (SYSID), Saint-Malo, France.(invited paper)*, volume 102, page 117, 2009.

[89] N. Kantas, A. Doucet, S. S. Singh, J. M. Maciejowski, and N. Chopin. On particle methods for parameter estimation in state-space models. *arXiv preprint arXiv:1412.8695*, 2014.

[90] M. Kilbinger, K. Benabed, O. Cappé, J. F. Cardoso, J. Coupon, G. Fort, H. J. McCracken, S. Prunet, C. P. Robert, and D. Wraith. CosmoPMC: Cosmology population Monte Carlo. *arXiv preprint arXiv:1101.0950*, 2011.

[91] M. Kilbinger, D. Wraith, C. P. Robert, K. Benabed, O. Cappé, J.F. Cardoso, G. Fort, S. Prunet, and F.R. Bouchet. Bayesian model comparison in cosmology with population Monte Carlo. *Monthly Notices of the Royal Astronomical Society*, 405(4):2381–2390, 2010.

[92] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25, 1996.

[93] G. Kitagawa. A self-organizing state-space model. *Journal of the American Statistical Association*, pages 1203–1215, 1998.

[94] E. Koblents and J. Míguez. A population Monte Carlo method for Bayesian inference and its application to stochastic kinetic models. In *EUSIPCO*, 2011.

[95] E. Koblents and J. Míguez. A population Monte Carlo scheme for computational inference in high dimensional spaces. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6318–6322. IEEE, 2013.

[96] E. Koblents and J. Míguez. A population Monte Carlo scheme with transformed weights and its application to stochastic kinetic models. *Statistics and Computing*, pages 1–19, 2013.

[97] E. Koblents and J. Míguez. Robust mixture population Monte Carlo scheme with adaptation of the number of components. In *EUSIPCO*, 2013.

[98] E. Koblents and J. Míguez. A comparison of nonlinear population Monte Carlo and particle Markov chain Monte Carlo algorithms for Bayesian inference in stochastic kinetic models. *arXiv preprint arXiv:1404.5218*, 2014.

[99] E. Koblents, J. Míguez, M. A. Rodríguez, and A. M. Schmidt. A nonlinear population Monte Carlo scheme for the Bayesian estimation of parameters of α-stable distributions. *Submitted to Computational Statistics and Data Analysis*, 2014.

[100] S. M. Kogon and D. B. Williams. Characteristic function based estimation of stable distribution parameters. *A Practical Guide to Heavy Tailed Data*, pages 311–335, 1998.

[101] A. Kong, J. S. Liu, and W. H. Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 9:278–288, 1994.

[102] J. H Kotecha and P. M. Djuric. Gaussian particle filtering. *Signal Processing, IEEE Transactions on*, 51(10):2592–2601, 2003.

[103] I. A. Koutrouvelis. An iterative procedure for the estimation of the parameters of stable laws: An iterative procedure for the estimation.

Communications in Statistics-Simulation and Computation, 10(1):17–28, 1981.

[104] E. E. Kuruoglu. Density parameter estimation of skewed α-stable distributions. *Signal Processing, IEEE Transactions on*, 49(10):2192–2201, 2001.

[105] A. Lee, C. Yau, M. B. Giles, A. Doucet, and C. C. Holmes. On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. *Journal of computational and graphical statistics*, 19(4):769–789, 2010.

[106] D. S. Lee and N. K. K. Chia. A particle algorithm for sequential Bayesian parameter estimation and model selection. *Signal Processing, IEEE Transactions on*, 50(2):326–336, 2002.

[107] J. E. Lee, R. McVinish, and K. Mengersen. Population Monte Carlo algorithm in high dimensions. *Methodology and Computing in Applied Probability*, 13(2):369–389, 2011.

[108] P. M. Lee. *Bayesian statistics: an introduction*. John Wiley & Sons, 2012.

[109] F. LeGland and L. Mevel. Recursive estimation in hidden Markov models. In *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, volume 4, pages 3468–3473. IEEE, 1997.

[110] C. E. Leith. Theoretical skill of Monte Carlo forecasts. *Monthly Weather Review*, 102(6):409–418, 1974.

[111] W. R. Leo. *Techniques for nuclear and particle physics experiments: a how-to approach*. Springer Science & Business Media, 2012.

[112] A. Lewis and S. Bridle. Cosmological parameters from cmb and other data: a Monte Carlo approach. *Phys. Rev.*, D66:103511, 2002.

[113] J. Liu and M. West. Combined parameter and state estimation in simulation-based filtering. In *Sequential Monte Carlo methods in practice*, pages 197–223. Springer, 2001.

[114] J. S. Liu. *Monte Carlo strategies in scientific computing*. Springer, 2008.

[115] J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044, 1998.

[116] J. S. Liu, F. Liang, and W. H. Wong. The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134, 2000.

[117] M. Ljungberg, S. E. Strand, and M. A. King. *Monte Carlo calculations in nuclear medicine: Applications in diagnostic imaging*. CRC Press, 2012.

[118] M. J. Lombardi. Bayesian inference for α-stable distributions: A random walk MCMC approach. *Computational Statistics & Data Analysis*, 51(5):2688–2700, 2007.

[119] H. T. MacGillivray, R. J. Dodd, B. V. McNally, J. F. Lightfoot, H. G. Corwin Jr., and S. R. Heathcote. Monte Carlo simulations of galaxy systems. *Astrophysics and Space Science*, 81(1-2):231–250, 1982.

[120] B. F. J. Manly. *Randomization, bootstrap and Monte Carlo methods in biology*, volume 70. CRC Press, 2006.

[121] E. Marinari and G. Parisi. Simulated tempering: a new Monte Carlo scheme. *EPL (Europhysics Letters)*, 19(6):451, 2007.

[122] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328, 2003.

[123] A .W. Marshall. The use of multi-stage sampling schemes in Monte Carlo computations. symposium on Monte Carlo methods, 123-140, edited by ma meyer, 1956.

[124] J. H. McCulloch. Simple consistent estimators of stable distribution parameters. *Communications in Statistics-Simulation and Computation*, 15(4):1109–1136, 1986.

[125] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[126] N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

[127] J. Míguez and E. Koblents. Particle filtering with transformed weights. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5th International Workshop on*, pages 364–367. IEEE, 2013.

[128] P. Milner, C.S. Gillespie, and D.J. Wilkinson. Moment closure based parameter inference of stochastic kinetic models. *Statistics and Computing*, pages 1–9, 2013.

[129] A. Mücke, R. Engel, J. P. Rachen, R. J. Protheroe, and T. Stanev. Monte Carlo simulations of photohadronic processes in astrophysics. *Computer Physics Communications*, 124(2):290–314, 2000.

[130] R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

[131] R. M. Neal. Slice sampling. *Annals of statistics*, pages 705–741, 2003.

[132] R. Neapolitan. *Learning Bayesian Networks*. Prentice-Hall, 2004.

[133] K. B. Newman, C. Fernández, L. Thomas, and S. T. Buckland. Monte Carlo inference for state–space models of wild animal populations. *Biometrics*, 65(2):572–583, 2009.

[134] C. L. Nikias and M. Shao. *Signal processing with alpha-stable distributions and applications*. Wiley-Interscience, 1995.

[135] J. P. Nolan. Numerical calculation of stable densities and distribution functions. *Communications in Statistics. Stochastic models*, 13(4):759–774, 1997.

[136] J. P. Nolan. Fitting data and assessing goodness-of-fit with stable distributions. *Unpublished Manuscript. Washington, DC: American University*, 1999.

[137] J. P. Nolan. Maximum likelihood estimation of stable parameters. In *Lévy processes: Theory and Applications*, pages 379–400. Boston: Birkhäuser, 2001.

[138] J. P. Nolan. *Stable Distributions - Models for Heavy Tailed Data*. Birkhäuser, Boston, 2013. In progress, Chapter 1 online at academic2.american.edu/∼jpnolan.

[139] A. Papavasiliou. Parameter estimation and asymptotic stability in stochastic filtering. *Stochastic processes and their applications*, 116(7):1048–1065, 2006.

[140] G. W. Peters, S. A. Sisson, and Y. Fan. Likelihood-free Bayesian inference for α-stable models. *Computational Statistics & Data Analysis*, 56(11):3743–3756, 2012.

[141] M. M. Pieri, H. Martel, and C. Grenon. Anisotropic galactic outflows and enrichment of the intergalactic medium. i. Monte Carlo simulations. *The Astrophysical Journal*, 658(1):36, 2007.

[142] M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.

[143] J. K. Pritchard, M. T. Seielstad, A. Perez-Lezaun, and M. W. Feldman. Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular Biology and Evolution*, 16(12):1791–1798, 1999.

[144] S. Rachev and S. Mittnik. Stable paretian models in finance. 2000.

[145] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*, volume 685. Artech house Boston, 2004.

[146] C. P. Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.

[147] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[148] D. W. O. Rogers. Fifty years of Monte Carlo simulations for medical physics. *Physics in medicine and biology*, 51(13):R287, 2006.

[149] B. Shen, M. F. Bugallo, and P. M. Djuric. Multiple marginalized population Monte Carlo. In *EUSIPCO*, 2010.

[150] B. Shen, M. F. Bugallo, and P. M. Djuric. Estimation of multimodal posterior distributions of chirp parameters with population Monte Carlo sampling. In *ICASSP*, pages 3861–3864. IEEE, 2012.

[151] S. A. Sisson, Y. Fan, and M. M. Tanaka. Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765, 2007.

[152] G. Storvik. Particle filters for state-space models with the presence of unknown static parameters. *Signal Processing, IEEE Transactions on*, 50(2):281–289, 2002.

[153] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial intelligence*, 128(1):99–141, 2001.

[154] R. Trotta. Applications of Bayesian model selection to cosmological parameters. *Monthly Notices of the Royal Astronomical Society*, 378(1):72–82, 2007.

[155] B. M. Turner and T. Van Zandt. A tutorial on approximate Bayesian computation. *Journal of Mathematical Psychology*, 56(2):69–85, 2012.

[156] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. Wan. The unscented particle filter. In *NIPS*, pages 584–590, 2000.

[157] J. Vermaak, C. Andrieu, A. Doucet, and S. J. Godsill. Particle methods for Bayesian modeling and enhancement of speech signals. *Speech and Audio Processing, IEEE Transactions on*, 10(3):173–185, 2002.

[158] V. Volterra. Fluctuations in the abundance of a species considered mathematically. *Nature*, 118:558–560, 1926.

[159] D. Vose. *Risk analysis: a quantitative guide*. John Wiley & Sons, 2008.

[160] D. J. Wilkinson. Parameter inference for stochastic kinetic models of bacterial gene regulation: A Bayesian approach to systems biology. *(with discussion), in Bayesian Statistics*, 9, 2011.

[161] D. J. Wilkinson. *Stochastic modelling for systems biology*, volume 44. CRC press, 2011.

[162] D. Williams. *Probability with Martingales*. Cambdridge University Press, Cambridge (UK), 1991.

[163] G. Winkler. *Image analysis, random fields and Markov chain Monte Carlo methods: a mathematical introduction*, volume 27. Springer, 2003.

[164] D. Wraith, M. Kilbinger, K. Benabed, O. Cappé, J. F. Cardoso, G. Fort, S. Prunet, and C. P. Robert. Estimation of cosmological parameters using adaptive importance sampling. *Physical Review D*, 80(2):023507, 2009.

[165] P. Zannetti. New Monte Carlo scheme for simulating Lagranian particle diffusion with wind shear effects. *Applied Mathematical Modelling*, 8(3):188–192, 1984.